ACM PORTAL DIGITAL LIBRARY

> home > about > feedback > login

US Patent & Trademark Office

Try the *new* Portal design

Give us your opinion after using it.

Search Results

Search Results for: **[3D<AND>((CAD and (hous* or building or architecture) and cost)) <AND>(meta_published_date <= 12-01-1997)]**
Found **246** of **571 searched out of 126,861.**

Warning: Maximum result set of 200 exceeded. Consider refining.

Search within Results

GO > Advanced Search

> Search Help/Tips

Sort by: Title Publication Publication Date Score

Results 1 - 20 of 200 **short listing**

Prev Page 1 2 3 4 5 6 7 8 9 10 Next Page

1 Virtual reality and simulation 90%
Martin Barnes
Proceedings of the 28th conference on Winter simulation November 1996

2 Fitting smooth surfaces to dense polygon meshes 87%
Venkat Krishnamurthy , Marc Levoy
Proceedings of the 23rd annual conference on Computer graphics and interactive techniques August 1996

3 Quest—queueing event simulation tool 85%
Joseph C. Hugan
Proceedings of the 26th conference on Winter simulation December 1994

4 Computer-Aided Design, Manufacturing, Assembly and Test (CADMAT) 85%
F. C. Bergsten
Proceedings of the eighteenth design automation conference on Design automation June 1981

This paper describes an integrated Computer-Aided Design, Manufacturing, Assembly and Test System to achieve increased productivity and quality. The term CADMAT which encompasses Automated Test also implies the integration of all design automated and computer augmented processes throughout the product and business organizations. In the design and test phase of electronics, Interactive Graphiss (IAG), verification, automated board layout, and automatic test vector generation are discussed. I ...

5 Networked VR system: kitchen layout design for customers 85%
Tomohiro Fukuda , Ryuichiro Nagahama , Junji Nomura
Proceedings f the second symposium on Virtual reality m deling language
February 1997

6 Model-based object recognition in dense-range images—a review 85%
Farshid Arman , J. K. Aggarwal
ACM Computing Surveys (CSUR) March 1993
Volume 25 Issue 1
The goal in computer vision systems is to analyze data collected from the environment and derive an interpretation to complete a specified task. Vision system tasks may be divided into data acquisition, low-level processing, representation, model construction, and matching subtasks. This paper presents a comprehensive survey of model-based vision systems using dense-range images. A comprehensive survey of the recent publications in each subtask pertaining to dense-range image object recogni ...

7 A human's eye view: motion blur and frameless rendering 84%
Ellen J. Scher Zagier
Crossroads May 1997
Volume 3 Issue 4

8 A data front-end for layered manufacturing 84%
Gill Barequet , Yuval Kaplan
Proceedings of the thirteenth annual symposium on Computational geometry
August 1997

9 Dimension-independent modeling with simplicial complexes 84%
A. Paoluzzi , F. Bernardini , C. Cattani , V. Ferrucci
ACM Transactions on Graphics (TOG) January 1993
Volume 12 Issue 1

10 Toward a history of (personal) workstations 84%
Gordon Bell
Proceedings of the ACM Conference on The history of personal workstations
January 1986
I originally accepted this keynote honor for five reasons: to respond to Alan Perlis' request (he told me I could present anything from a new taxonomy to personal reminiscences); second, to identify the important artifacts that should be preserved in The Computer Museum; third, to posit a framework of the history of workstations that can be written in the next century (we're all too close to create it); fourth, to summarize my own involvement on interactive computing including timesharing a ...

11 Power macromodeling for high level power estimation 83%
Subodh Gupta , Farid N. Najm
Proceedings of the 34th annual conference on Design automation conference
June 1997

12 Rapid design and prototyping of customized rehabilitation aids 83%
Vijay Kumar , Ruzena Bajcsy , William Harwin , Patrick Harker
Communications of the ACM February 1996

Volume 39 Issue 2

13 The triangle processor and normal vector shader: a VLSI system for high performance graphics 82%
Michael Deering , Stephanie Winner , Bic Schediwy , Chris Duffy , Neil Hunt
ACM SIGGRAPH Computer Graphics , Proceedings of the 15th annual conference on Computer graphics and interactive techniques June 1988
Volume 22 Issue 4

14 The future of simulation software: a panel discussion 82%
Jerry Banks
Proceedings of the 29th conference on Winter simulation December 1997

15 Machine interpretation of CAD data for manufacturing applications 82%
Qiang Ji , Michael M. Marefat
ACM Computing Surveys (CSUR) September 1997
Volume 29 Issue 3
Machine interpretation of the shape of a component for CAD databases is an important problem in CAD/CAM, computer vision, and intelligent manufacturing. It can be used in CAD/CAM for evaluation of designs, in computer vision for machine recognition and machine inspection of objects, and in intelligent manufacturing for automating and integrating the link between design and manufacturing. This topic has been an active area of research since the late '70s, and a significant number of computat ...

16 Visibility culling using hierarchical occlusion maps 82%
Hansong Zhang , Dinesh Manocha , Tom Hudson , Kenneth E. Hoff
Proceedings of the 24th annual conference on Computer graphics and interactive techniques August 1997

17 Talisman: commodity realtime 3D graphics for the PC 82%
Jay Torborg , James T. Kajiya
Proceedings of the 23rd annual conference on Computer graphics and interactive techniques August 1996

18 OBBTree: a hierarchical structure for rapid interference detection 82%
S. Gottschalk , M. C. Lin , D. Manocha
Proceedings of the 23rd annual conference on Computer graphics and interactive techniques August 1996

19 Computer interactive sculpture 82%
Helaman Ferguson
Proceedings of the 1992 symposium on Interactive 3D graphics June 1992

20 Gross motion planning—a survey 82%
Yong K. Hwang , Narendra Ahuja
ACM Computing Surveys (CSUR) September 1992
Volume 24 Issue 3
Motion planning is one of the most important areas of robotics research. The complexity of the motion-planning problem has hindered the development of practical

algorithms. This paper surveys the work on gross-motion planning, including motion planners for point robots, rigid robots, and manipulators in stationary, time-varying, constrained, and movable-object environments. The general issues in motion planning are explained. Recent approaches and their performances are briefly described, a ...

Results 1 - 20 of 200 **short listing**

Prev Page 1 2 3 4 5 6 7 8 9 10 Next Page

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

PORTAL
THE ACM DIGITAL LIBRARY

> home > about > feedback > login
US Patent & Trademark Office

Try the *new* Portal design
Give us your opinion after using it.

Search Results

Search Results for: **[polyhedron and (resize or recalculate or redimension)]**
Found **29** of **126,861 searched.**

Search within Results

GO > Advanced Search

> Search Help/Tips

Sort by: **Title** **Publication** **Publication Date** **Score** Binder

Results 1 - 20 of 29 **short listing**

Prev Page 1 2 Next Page

1 Folding flat silhouettes and wrapping polyhedral packages: new results in computational origami 85%
Erik D. Demaine , Martin L. Demaine , Joseph S. B. Mitchell
Proceedings of the fifteenth annual symposium on Computational geometry June 1999

2 A constraint-based manipulator toolset for editing 3D objects 82%
C. Hsu , G. Alt , Z. Huang , E. Beier , B. Brüderlin
Proceedings of the fourth ACM symposium on Solid modeling and applications May 1997

3 Hierarchical view-dependent structures for interactive scene manipulation 82%
Normand Brière , Pierre Poulin
Proceedings of the 23rd annual conference on Computer graphics and interactive techniques August 1996

4 A parametric surface blending method for complex engineering objects 82%
T. Lee , S. Bedi , R. N. Dubey
Proceedings on the second ACM symposium on Solid modeling and applications June 1993

5 Free-form deformations with lattices of arbitrary topology 80%
Ron MacCracken , Kenneth I. Joy
Proceedings of the 23rd annual conference on Computer graphics and interactive techniques August 1996

6 Visibility sorting and compositing without splitting for image layer decompositions 80%

John Snyder , Jed Lengyel

Proceedings of the 25th annual conference on Computer graphics and interactive techniques July 1998

7 Rendering with coherent layers 80%

Jed Lengyel , John Snyder

Proceedings of the 24th annual conference on Computer graphics and interactive techniques August 1997

8 Polygon shadow generation 77%

Peter Atherton , Kevin Weiler , Donald Greenberg

ACM SIGGRAPH Computer Graphics , Proceedings of the 5th annual conference on Computer graphics and interactive techniques August 1978

Volume 12 Issue 3

A general purpose method for generating shadows using a polygonal coordinate data base is presented. The method is based on an object space polygon clipping hidden surface removal algorithm. Output from the program is in the same three-dimensional polygon format as the input. Thus, a shadowed data environment may be easily created and viewed from any observer position with no additional depth sorting time required for the hidden surface removal process. Shadows can also be cast by more than ...

9 Replication strategies in unstructured peer-to-peer networks 77%

Edith Cohen , Scott Shenker

ACM SIGCOMM Computer Communication Review , Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications August 2002

Volume 31 Issue 4

The Peer-to-Peer (P2P) architectures that are most prevalent in today's Internet are decentralized and unstructured. Search is blind in that it is independent of the query and is thus not more effective than probing randomly chosen peers. One technique to improve the effectiveness of blind search is to proactively replicate data. We evaluate and compare different replication strategies and reveal interesting structure: Two very common but very different replication strategies - uniform and propo ...

10 Algorithms and applications for answering ranked queries using ranked views 77%

Vagelis Hristidis , Yannis Papakonstantinou

The VLDB Journal — The International Journal on Very Large Data Bases January 2004

Volume 13 Issue 1

Ranked queries return the top objects of a database according to a preference function. We present and evaluate (experimentally and theoretically) a core algorithm that answers ranked queries in an efficient pipelined manner using materialized ranked views. We use and extend the core algorithm in the described PREFER and MERGE systems. PREFER precomputes a set of materialized views that provide guaranteed query performance. We present an algorithm that selects a near optimal set of views under s ...

11 Automation of re-entry vehicles' preliminary design 77%

John M. Jeglic , Roy F. Sewall , C. David Magrath

Proceedings of the 6th annual conference on Design Automation January 1969

The science of computer graphics is gaining undisputed momentum in the rapidly growing computer field. While it is true that automated drafting has had a measurable effect on design technology, particularly in the reduction of costs, one cannot classify it as automated design. Automated drafting simply permits the designer to automate the final phase of his work and to either increase the quantity, and possibly the quality, of the final output of drawings, or to give him more time for creat ...

12 Interval Set: A Volume Rendering Technique Generalizing Isosurface Extraction 77%

Baining Guo

Proceedings of the 6th conference on Visualization '95 October 1995

A scalar volume V = { (x, f(x)) | x in R } is described by a function f(x) defined over some region R of the three-dimensional space. This paper presents a simple technique for rendering interval sets of the form I(a, b) = { (x, f(x)) | a <= g(x) <= b }, where a and b are either real numbers or infinities. We describe an algorithm for triangulating interval sets as alpha shapes, which can be accurately and efficiently rendered as surfaces or semi-transparent clouds. On the theoretical side, inte ...

13 Session P9: view-dependent visualization: Visibility-guided simplification 77%

Eugene Zhang , Greg Turk

Proceedings of the conference on Visualization '02 October 2002

For some graphics applications, object interiors and hard-to-see regions contribute little to the final images and need not be processed. In this paper, we define a view-independent visibility measure on mesh surfaces based on the visibility function between the surfaces and a surrounding sphere of cameras. We demonstrate the usefulness of this measure with a visibility-guided simplification algorithm.Mesh simplification reduces the polygon counts of 3D models and speeds up the rendering process ...

14 Poster Session: Cutting 3D freeform objects with genus-n into single boundary surfaces using topological graphs 77%

Dvir Steiner , Anath Fischer

Proceedings of the seventh ACM symposium on Solid modeling and applications June 2002

In reverse engineering, surface reconstruction methods for freeform objects are based mainly on geometrical criteria, while topological factors are neglected. Current methods use a bottom-up approach based on local parameterization to reconstruct the object from points to a dense mesh and finally to smooth connected patches. This type of reconstruction, however, can have topological problems that might lead to parameterization difficulties, noisy surface behavior and texture anomalies. Such prob ...

15 Color gamut mapping and the printing of digital color images 77%

Maureen C. Stone , William B. Cowan , John C. Beatty

ACM Transactions on Graphics (TOG) October 1988

Volume 7 Issue 4

Principles and techniques useful for calibrated color reproduction are defined. These results are derived from a project to take digital images designed on a variety of different color monitors and accurately reproduce them in a journal using digital offset printing. Most of the images printed were reproduced without access to the image as

viewed in its original form; the color specification was derived entirely from calorimetric specification. The techniques described here are not specific ...

16 A shading model for atmospheric scattering considering luminous intensity distribution of light sources 77%

Tomoyuki Nishita , Yasuhiro Miyawaki , Eihachiro Nakamae

ACM SIGGRAPH Computer Graphics , Proceedings of the 14th annual conference on Computer graphics and interactive techniques August 1987

Volume 21 Issue 4

17 Optimizing cache miss equations polyhedra 77%

Nerina Bermudo , Xavier Vera , Antonio González , Josep Llosa

ACM SIGARCH Computer Architecture News March 2000

Volume 28 Issue 1

Cache Miss Equations (CME) [GMM97] is a method that accurately describes the cache behavior by means of polyhedra. Even though the computation cost of generating CME is a linear function of the number of references, to solve them is a very time consuming task and thus trying to study a whole program may be infeasible.In this work, we present effective techniques that exploit some properties of the particular polyhedra generated by CME. Such technique reduce the complexity of the algorithm to sol ...

18 Integrated analytic spatial and temporal anti-aliasing for polyhedra in 4-space 77%

Charles W. Grant

ACM SIGGRAPH Computer Graphics , Proceedings of the 12th annual conference on Computer graphics and interactive techniques July 1985

Volume 19 Issue 3

19 New type signatures for legacy Fortran subroutines 77%

Nicky Williams Preston

ACM SIGSOFT Software Engineering Notes , Proceedings of the 1999 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering September 1999

Volume 24 Issue 5

We are currently developing a methodological framework for reverse engineering Fortran77 programs used by Electricité de France, in which the first step is the construction of an algebraic specification which faithfully represents the Fortran code. To construct this specification, we must decide on a coherent set of "profiles" (type signatures) for the specifications of the Fortran sub-programs. We propose an analysis of the dynamic aliases between formal and actual sub-program argume ...

20 A coherent sweep plane slicer for layered manufacturing 77%

Sara McMains , Carlo Séquin

Proceedings of the fifth ACM symposium on Solid modeling and applications June 1999

Results 1 - 20 of 29 **short listing**

Prev Page 1 2 Next Page

PORTAL
THE ACM DIGITAL LIBRARY

> home > about > feedback > login

US Patent & Trademark Office

Try the *new* Portal design
Give us your opinion after using it.

Search Results

Search Results for: **[polyhedron and (resize or recalculate or redimension)]**
Found **29** of **126,861 searched.**

Search within Results

GO > Advanced Search

> Search Help/Tips

Sort by: **Title** **Publication** **Publication Date** **Score** Binder

Results 21 - 29 of 29 **short listing**

Prev Page 1 2 Next Page

21 Graphical modeling and animation of brittle fracture 77%
James F. O'Brien , Jessica K. Hodgins
Proceedings of the 26th annual conference on Computer graphics and interactive techniques July 1999

22 Surface construction from within a virtual environment 77%
Andrew Pieter Van Pernis
Proceedings of the 37th annual Southeast regional conference (CD-ROM) April 1999

23 Minkowski operations for satellite antenna layout 77%
Jean-Daniel Boissonnat , Eelco de Lange , Monique Teillaud
Proceedings of the thirteenth annual symposium on Computational geometry August 1997

24 Position-based physics: simulating the motion of many highly interacting spheres and polyhedra 77%
Victor J. Milenkovic
Proceedings of the 23rd annual conference on Computer graphics and interactive techniques August 1996

25 Bubble mesh: automated triangular meshing of non-manifold geometry by sphere packing 77%
Kenji Shimada , David C. Gossard
Proceedings of the third ACM symposium n Solid modeling and applications December 1995

26 Inexpensive advanced graphics applications for the C.S. majors graphics class 77%

Lee H. Tichenor

ACM SIGCSE Bulletin , Pr ceedings of the twenty-sixth SIGCSE technical symposium on Computer science education March 1995

Volume 27 Issue 1

To demonstrate more advanced topics we have changed the focus of our graphics assignments in our upper division C.S. majors course from strictly programming basic algorithms to working in existing software packages. In addition to three standard programming assignments in Pascal or C the students perform experiments and develop designs with L-grammar, fractal generator, morph, ray-tracing, and animation packages. Excellent and inexpensive versions of all these systems are available through ...

27 I-COLLIDE: an interactive and exact collision detection system for large-scale environments 77%

Jonathan D. Cohen , Ming C. Lin , Dinesh Manocha , Madhav Ponamgi

Proceedings of the 1995 symposium on Interactive 3D graphics April 1995

we present an exact and interactive collision detection system, I-COLLIDE, for large-scale environments. Such environments are characterized by the number of objects undergoing rigid motion and the complexity of the models. The algorithm does not assume the objects' motions can be expressed as a closed form function of time. The collision detection system is general and can be easily interfaced with a variety of applications. The algorithm uses a two-level approach based on pruning multiple ...

28 Fast object-precision shadow generation for area light source 77%

Norman Chin , Steven Feiner

Proceedings of the 1992 symposium on Interactive 3D graphics June 1992

29 Model-based object recognition in dense-range images—a review 77%

Farshid Arman , J. K. Aggarwal

ACM Computing Surveys (CSUR) March 1993

Volume 25 Issue 1

The goal in computer vision systems is to analyze data collected from the environment and derive an interpretation to complete a specified task. Vision system tasks may be divided into data acquisition, low-level processing, representation, model construction, and matching subtasks. This paper presents a comprehensive survey of model-based vision systems using dense-range images. A comprehensive survey of the recent publications in each subtask pertaining to dense-range image object recogni ...

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

PORTAL
THE ACM DIGITAL LIBRARY

> home > about > feedback > login

US Patent & Trademark Office

Try the *new* Portal design
Give us your opinion after using it.

Search Results

Search Results for: **[area<AND>((polyhedron and (resize or recalculate or redimension)))]**
Found **19** of **126,861 searched.**

Search within Results

GO > Advanced Search

> Search Help/Tips

Sort by: **Title** **Publication** **Publication Date** **Score** Binder

Results 1 - 19 of 19 **short listing**

1 Folding flat silhouettes and wrapping polyhedral packages: new results in computational origami 85%
Erik D. Demaine , Martin L. Demaine , Joseph S. B. Mitchell
Proceedings of the fifteenth annual symposium on Computational geometry June 1999

2 A parametric surface blending method for complex engineering objects 82%
T. Lee , S. Bedi , R. N. Dubey
Proceedings on the second ACM symposium on Solid modeling and applications June 1993

3 Free-form deformations with lattices of arbitrary topology 80%
Ron MacCracken , Kenneth I. Joy
Proceedings of the 23rd annual conference on Computer graphics and interactive techniques August 1996

4 Hierarchical view-dependent structures for interactive scene manipulation 80%
Normand Brière , Pierre Poulin
Proceedings of the 23rd annual conference on Computer graphics and interactive techniques August 1996

5 Rendering with coherent layers 80%
Jed Lengyel , John Snyder
Proceedings of the 24th annual conference on Computer graphics and interactive techniques August 1997

6 Polygon shadow generation 77%

Peter Atherton , Kevin Weiler , Donald Greenberg

ACM SIGGRAPH Computer Graphics , Proceedings of the 5th annual conference on C mputer graphics and interactive techniques August 1978

Volume 12 Issue 3

A general purpose method for generating shadows using a polygonal coordinate data base is presented. The method is based on an object space polygon clipping hidden surface removal algorithm. Output from the program is in the same three-dimensional polygon format as the input. Thus, a shadowed data environment may be easily created and viewed from any observer position with no additional depth sorting time required for the hidden surface removal process. Shadows can also be cast by more than ...

7 Replication strategies in unstructured peer-to-peer networks 77%

Edith Cohen , Scott Shenker

ACM SIGCOMM Computer Communication Review , Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications August 2002

Volume 31 Issue 4

The Peer-to-Peer (P2P) architectures that are most prevalent in today's Internet are decentralized and unstructured. Search is blind in that it is independent of the query and is thus not more effective than probing randomly chosen peers. One technique to improve the effectiveness of blind search is to proactively replicate data. We evaluate and compare different replication strategies and reveal interesting structure: Two very common but very different replication strategies - uniform and propo ...

8 Algorithms and applications for answering ranked queries using ranked views 77%

Vagelis Hristidis , Yannis Papakonstantinou

The VLDB Journal — The International Journal on Very Large Data Bases January 2004

Volume 13 Issue 1

Ranked queries return the top objects of a database according to a preference function. We present and evaluate (experimentally and theoretically) a core algorithm that answers ranked queries in an efficient pipelined manner using materialized ranked views. We use and extend the core algorithm in the described PREFER and MERGE systems. PREFER precomputes a set of materialized views that provide guaranteed query performance. We present an algorithm that selects a near optimal set of views under s ...

9 Interval Set: A Volume Rendering Technique Generalizing Isosurface Extraction 77%

Baining Guo

Proceedings of the 6th conference on Visualization '95 October 1995

A scalar volume V = { (x, f(x)) | x in R } is described by a function f(x) defined over some region R of the three-dimensional space. This paper presents a simple technique for rendering interval sets of the form I(a, b) = { (x, f(x)) | a <= g(x) <= b }, where a and b are either real numbers or infinities. We describe an algorithm for triangulating interval sets as alpha shapes, which can be accurately and efficiently rendered as surfaces or semi-transparent clouds. On the theoretical side, inte ...

10 Session P9: view-dependent visualization: Visibility-guided simplification 77%

Eugene Zhang , Greg Turk

Proceedings f the c nference n Visualization '02 October 2002

For some graphics applications, object interiors and hard-to-see regions contribute little to the final images and need not be processed. In this paper, we define a view-independent visibility measure on mesh surfaces based on the visibility function between the surfaces and a surrounding sphere of cameras. We demonstrate the usefulness of this measure with a visibility-guided simplification algorithm.Mesh simplification reduces the polygon counts of 3D models and speeds up the rendering process ...

11 Poster Session: Cutting 3D freeform objects with genus-n into single boundary surfaces using topological graphs 77%

Dvir Steiner , Anath Fischer

Proceedings of the seventh ACM symposium on Solid modeling and applications June 2002

In reverse engineering, surface reconstruction methods for freeform objects are based mainly on geometrical criteria, while topological factors are neglected. Current methods use a bottom-up approach based on local parameterization to reconstruct the object from points to a dense mesh and finally to smooth connected patches. This type of reconstruction, however, can have topological problems that might lead to parameterization difficulties, noisy surface behavior and texture anomalies. Such prob ...

12 Color gamut mapping and the printing of digital color images 77%

Maureen C. Stone , William B. Cowan , John C. Beatty

ACM Transactions on Graphics (TOG) October 1988

Volume 7 Issue 4

Principles and techniques useful for calibrated color reproduction are defined. These results are derived from a project to take digital images designed on a variety of different color monitors and accurately reproduce them in a journal using digital offset printing. Most of the images printed were reproduced without access to the image as viewed in its original form; the color specification was derived entirely from calorimetric specification. The techniques described here are not specific ...

13 A shading model for atmospheric scattering considering luminous intensity distribution of light sources 77%

Tomoyuki Nishita , Yasuhiro Miyawaki , Eihachiro Nakamae

ACM SIGGRAPH Computer Graphics , Proceedings of the 14th annual conference on Computer graphics and interactive techniques August 1987

Volume 21 Issue 4

14 Integrated analytic spatial and temporal anti-aliasing for polyhedra in 4-space 77%

Charles W. Grant

ACM SIGGRAPH Computer Graphics , Proceedings of the 12th annual conference on Computer graphics and interactive techniques July 1985

Volume 19 Issue 3

15 A coherent sweep plane slicer for layered manufacturing 77%

Sara McMains , Carlo Séquin

Proceedings of the fifth ACM symposium on Solid modeling and applicati ns June 1999

16 Graphical modeling and animation of brittle fracture 77%

James F. O'Brien , Jessica K. Hodgins

Proceedings of the 26th annual conference on Computer graphics and interactive techniques July 1999

17 Inexpensive advanced graphics applications for the C.S. majors graphics class 77%

Lee H. Tichenor

ACM SIGCSE Bulletin , Proceedings of the twenty-sixth SIGCSE technical symposium on Computer science education March 1995

Volume 27 Issue 1

To demonstrate more advanced topics we have changed the focus of our graphics assignments in our upper division C.S. majors course from strictly programming basic algorithms to working in existing software packages. In addition to three standard programming assignments in Pascal or C the students perform experiments and develop designs with L-grammar, fractal generator, morph, ray-tracing, and animation packages. Excellent and inexpensive versions of all these systems are available through ...

18 Fast object-precision shadow generation for area light source 77%

Norman Chin , Steven Feiner

Proceedings of the 1992 symposium on Interactive 3D graphics June 1992

19 Model-based object recognition in dense-range images—a review 77%

Farshid Arman , J. K. Aggarwal

ACM Computing Surveys (CSUR) March 1993

Volume 25 Issue 1

The goal in computer vision systems is to analyze data collected from the environment and derive an interpretation to complete a specified task. Vision system tasks may be divided into data acquisition, low-level processing, representation, model construction, and matching subtasks. This paper presents a comprehensive survey of model-based vision systems using dense-range images. A comprehensive survey of the recent publications in each subtask pertaining to dense-range image object recogni ...

Results 1 - 19 of 19 short listing

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore® RELEASE 1.6

Welcome
United States Patent and Trademark Office

Help FAQ Terms IEEE Peer Review Quick Links » Se

Welcome to IEEE *Xplore*®

- Home
- What Can I Access?
- Log-out

Tables of Contents

- Journals & Magazines
- Conference Proceedings
- Standards

Search

- By Author
- Basic
- Advanced

Member Services

- Join IEEE
- Establish IEEE Web Account
- Access the IEEE Member Digital Library

Your search matched **0** of **1000582** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enter new one in the text box.

(resize or redimension or recalculate) and polyhedro Search

Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

Results:
No documents matched your query.

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ | Terms | Back to Top

Copyright © 2004 IEEE — All rights reserved

PORTAL DIGITAL LIBRARY

> home > about > feedback > login

US Patent & Trademark Office

Try the *new* Portal design

Give us your opinion after using it.

Search Results

Search Results for: **[3D<AND>((CAD and architecture and cost)) <AND> (meta_published_date <= 01-01-1998)]**
Found **145** of **353 searched out of 126,861.**

Search within Results

GO > Advanced Search

> Search Help/Tips

S rt by: Title Publication Publication Date Score Binder

Results 1 - 20 of 145 short listing

Prev Page 1 2 3 4 5 6 7 8 Next Page

1 Model-based object recognition in dense-range images—a review 85%

Farshid Arman , J. K. Aggarwal
ACM Computing Surveys (CSUR) March 1993
Volume 25 Issue 1

The goal in computer vision systems is to analyze data collected from the environment and derive an interpretation to complete a specified task. Vision system tasks may be divided into data acquisition, low-level processing, representation, model construction, and matching subtasks. This paper presents a comprehensive survey of model-based vision systems using dense-range images. A comprehensive survey of the recent publications in each subtask pertaining to dense-range image object recogni ...

2 Toward a history of (personal) workstations 84%

Gordon Bell
Proceedings of the ACM Conference on The history of personal workstations January 1986

I originally accepted this keynote honor for five reasons: to respond to Alan Perlis' request (he told me I could present anything from a new taxonomy to personal reminiscences); second, to identify the important artifacts that should be preserved in The Computer Museum; third, to posit a framework of the history of workstations that can be written in the next century (we're all too close to create it); fourth, to summarize my own involvement on interactive computing including timesharing a ...

3 The triangle processor and normal vector shader: a VLSI system for high performance graphics 82%

Michael Deering , Stephanie Winner , Bic Schediwy , Chris Duffy , Neil Hunt
ACM SIGGRAPH Computer Graphics , Proceedings of the 15th annual conference on Computer graphics and interactive techniques June 1988

Volume 22 Issue 4

4 Machine interpretation of CAD data for manufacturing applications 82%
Qiang Ji , Michael M. Marefat
ACM Computing Surveys (CSUR) September 1997
Volume 29 Issue 3
Machine interpretation of the shape of a component for CAD databases is an important problem in CAD/CAM, computer vision, and intelligent manufacturing. It can be used in CAD/CAM for evaluation of designs, in computer vision for machine recognition and machine inspection of objects, and in intelligent manufacturing for automating and integrating the link between design and manufacturing. This topic has been an active area of research since the late '70s, and a significant number of computat ...

5 Visibility culling using hierarchical occlusion maps 82%
Hansong Zhang , Dinesh Manocha , Tom Hudson , Kenneth E. Hoff
Proceedings of the 24th annual conference on Computer graphics and interactive techniques August 1997

6 Talisman: commodity realtime 3D graphics for the PC 82%
Jay Torborg , James T. Kajiya
Proceedings of the 23rd annual conference on Computer graphics and interactive techniques August 1996

7 Power macromodeling for high level power estimation 82%
Subodh Gupta , Farid N. Najm
Proceedings of the 34th annual conference on Design automation conference June 1997

8 Dimension-independent modeling with simplicial complexes 82%
A. Paoluzzi , F. Bernardini , C. Cattani , V. Ferrucci
ACM Transactions on Graphics (TOG) January 1993
Volume 12 Issue 1

9 Remote programmability of graphic interactions in a host/satellite configuration 80%
Samuel D. Moulton , Philip J. Corman
ACM SIGGRAPH Computer Graphics , Proceedings of the 3rd annual conference on Computer graphics and interactive techniques July 1976
Volume 10 Issue 2
This paper describes the remote programmability of graphic-oriented interactions in a large, time-shared host/multiple-satellite configuration. The philosophy and architecture of a pseudo machine -- the Programmable Graphics Processor (PGP) -- is presented along with a description of the higher-level language used in the host to specify programs for execution in the satellite. These programs are first compiled into PGP machine language in the host and then transmitted to the satellite where they ...

10 Pictorial man-machine communication 80%
W. Haas , M. Mrva , W. Tengler
Proceedings of the ACM '81 conference January 1981
The importance of computer graphics and image processing lies in the opportunities

they provide for the use of pictorial information in communicating with computer systems /11/. Pictorial communication can replace the traditional alphanumeric man-machine communication because it represents a very natural way of interfacing man and machine. In this way pictorial communication promises foreseeable advances in communication technology because the volume of data that can be collected and dissem ...

11 The emerging technology of CAD/CAM 80%



Larry Lichten

Proceedings of the 1984 annual conference of the ACM on The fifth generation challenge January 1984

Computer-Aided Design and Manufacture (CAD/CAM) represents a merging of technological advances in computer hardware and software with pressing needs in manufacturing industries. Integrated manufacturing systems--from computer graphics-aided design through engineering analysis and automated fabrication--are only now beginning to fulfill nearly twenty-five-year old promises of increased production efficiency. This paper summarizes CAD/CAM's evolution and its current state and then describes s ...

12 An efficient 3-D visualization technique for finite element models and other coarse volumes 80%



R. S. Gallagher , J. C. Nagtegaal

ACM SIGGRAPH Computer Graphics , Proceedings of the 16th annual conference on Computer graphics and interactive techniques July 1989

Volume 23 Issue 3

13 Layout-driven RTL binding techniques for high-level synthesis using accurate estimators 80%

Min Xu , Fadi J. Kurdahi

ACM Transactions on Design Automation of Electronic Systems (TODAES) October 1997

Volume 2 Issue 4

The importance of effective and efficient accounting of layout effects is well established in High-Level Synthesis (HLS), since it allows more realistic exploration of the design space and the generation of solutions with predictable metrics. This feature is highly desirable in order to avoid unnecessary iterations through the design process. In this article, we address the problem of layout-driven register-transfer-level (RTL) binding as this step has a direct relevance to the final perfor ...

14 Visualization of construction planning information 80%



Kathleen McKinney , John Kunz , Martin Fischer

Proceedings of the 3rd international conference on Intelligent user interfaces January 1997

15 Interactive feature extraction for a form feature conversion system 80%



Yong Seok Suh , Michael J. Wozny

Proceedings of the fourth ACM symposium on Solid modeling and applications May 1997

16 Optimized geometry compression for real-time rendering 80%



Mike M. Chow

Proceedings f the 8th c nference n Visualization '97 October 1997

17 Searching requirements for a system to support cooperative concept design in product development 80%
Tuomo Tuikka
Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques August 1997

18 Retrofitting fine art and design education in the age of computer technology 80%
Gregory Patrick Garvey
ACM SIGGRAPH Computer Graphics August 1997
Volume 31 Issue 3

19 Dynamic objects for engineering automation 80%
Robert E. Phillips
Communications of the ACM May 1997
Volume 40 Issue 5

20 V-COLLIDE: accelerated collision detection for VRML 80%
Thomas C. Hudson , Ming C. Lin , Jonathan Cohen , Stefan Gottschalk , Dinesh Manocha
Proceedings of the second symposium on Virtual reality modeling language
February 1997

Results 1 - 20 of 145 **short listing**

Prev Page **1 2 3 4 5 6 7 8** Next Page

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

PORTAL
THE ACM DIGITAL LIBRARY

> home > about > feedback > login

US Patent & Trademark Office

Try the *new* Portal design
Give us your opinion after using it.

Search Results

Search Results for: **[3D<AND>((CAD and architecture and cost)) <AND> (meta_published_date <= 01-01-1998)]**
Found **145** of **126,861 searched.**

Search within Results

GO > Advanced Search

> Search Help/Tips

Sort by: **Title** **Publication** **Publication Date** **Score** Binder

Results 21 - 40 of 145 **short listing**

Prev Page 1 2 3 4 5 6 7 8 Next Page

21 Accelerated walkthrough of large spline models 80%
Subodh Kumar , Dinesh Manocha , Hansong Zhang , Kenneth E. Hoff
Proceedings of the 1997 symposium on Interactive 3D graphics April 1997

22 Virtual reality in scientific visualization 80%
Steve Bryson
Communications of the ACM May 1996
Volume 39 Issue 5

23 A comprehensive estimation technique for high-level synthesis 80%
Seong Y. Ohm , Fadi J. Kurdahi , Nikil Dutt , Min Xu
Proceedings of the 8th international symposium on System synthesis September 1995

Abstract: We present an integrated approach aimed at predicting layout area needed to implement a behavioral description for a given performance goal. Our approach is novel because: (1) it accounts for all types of RT level components (FUs, buses, registers), (2) it is highly flexible, allowing the designer to tradeoff one type of resource with another and considers dependencies between these different types, (3) it is vertically integrated to include provably accurate physical level estimators, ...

24 Speeding up irregular applications in shared-memory multiprocessors: memory binding and group prefetching 80%
Zheng Zhang , Josep Torrellas
ACM SIGARCH Computer Architecture News , Proceedings of the 22nd annual international symposium on Computer architecture May 1995
Volume 23 Issue 2

While many parallel applications exhibit good spatial locality, other important codes in areas like graph problem-solving or CAD do not. Often, these irregular codes contain small records accessed via pointers. Consequently, while the former applications benefit from long cache lines, the latter prefer short lines. One good solution is to combine short lines with prefetching. In this way, each application can exploit the amount of spatial locality that it has. However, prefetching, if provided, ...

25 Pen computing: a technology overview and a vision 80%



André Meyer
ACM SIGCHI Bulletin July 1995
Volume 27 Issue 3

This work gives an overview of a new technology that is attracting growing interest in public as well as in the computer industry itself. The visible difference from other technologies is in the use of a pen or pencil as the primary means of interaction between a user and a machine, picking up the familiar pen and paper interface metaphor. From this follows a set of consequences that will be analyzed and put into context with other emerging technologies and visions.Starting with a short historic ...

26 Conceptual graphs in constraint based re-design 80%



O. W. Salomons , F. van Slooten , F. J. A. M. van Houten , H. J. J. Kals
Proceedings of the third ACM symposium on Solid modeling and applications December 1995

27 Geometric programming: a programming approach to geometric design 80%



Alberto Paoluzzi , Valerio Pascucci , Michele Vicentino
ACM Transactions on Graphics (TOG) July 1995
Volume 14 Issue 3

This article presents a functional programming approach to geometric design with embedded polyhedral complexes. Its main goals are to show the expressive power of the language as well as its usefulness for geometric design. The language, named PLASM (the Programming LAnguage for Solid Modeling), introduces a very high level approach to "constructive" or "generative" modeling. Geometrical objects are generated by evaluating some suitable language expressions. Because ...

28 HoloSketch: a virtual reality sketching/animation tool 80%



Michael F. Deering
ACM Transactions on Computer-Human Interaction (TOCHI) September 1995
Volume 2 Issue 3

This article describes HoloSketch, a virtual reality-based 3D geometry creation and manipulation tool. HoloSketch is aimed at providing nonprogrammers with an easy-to-use 3D "What-You-See-Is-What-You-Get" environment. Using head-tracked stereo shutter glasses and a desktop CRT display configuration, virtual objects can be created with a 3D wand manipulator directly in front of the user, at very high accuracy and much more rapidly than with traditional 3D drawing systems. HoloSke ...

29 Quest—queueing event simulation tool 80%



Joseph C. Hugan
Proceedings of the 26th conference on Winter simulation December 1994

30 I-COLLIDE: an interactive and exact collision detection system for large-scale environments 80%



Jonathan D. Cohen , Ming C. Lin , Dinesh Manocha , Madhav Ponamgi
Proceedings f the 1995 symp sium n Interactive 3D graphics April 1995
we present an exact and interactive collision detection system, I-COLLIDE, for large-scale environments. Such environments are characterized by the number of objects undergoing rigid motion and the complexity of the models. The algorithm does not assume the objects' motions can be expressed as a closed form function of time. The collision detection system is general and can be easily interfaced with a variety of applications. The algorithm uses a two-level approach based on pruning multiple ...

31 Integrating computer technology, people technology and application technology: strategies and case studies from Georgia Tech's Graphics, Visualization and Usability Center 80%
Jim Foley
Proceedings of the workshop on Advanced visual interfaces June 1994

32 Radioptimization: goal based rendering 80%
John K. Kawai , James S. Painter , Michael F. Cohen
Proceedings of the 20th annual conference on Computer graphics and interactive techniques September 1993

33 Computational strategies for object recognition 80%
Paul Suetens , Pascal Fua , Andrew J. Hanson
ACM Computing Surveys (CSUR) March 1992
Volume 24 Issue 1
This article reviews the available methods for automated identification of objects in digital images. The techniques are classified into groups according to the nature of the computational strategy used. Four classes are proposed: (1) the simplest strategies, which work on data appropriate for feature vector classification, (2) methods that match models to symbolic data structures for situations involving reliable data and complex models, (3) approaches that fit models to the photometry and ...

34 Computer-aided visualization of the human body for industrial designer 80%
Alan Wier
Proceedings of the 18th annual ACM SIGUCCS conference on User services August 1990

35 Information technologies for the 1990s: an organizational impact perspective 80%
Detmar W. Straub , James C. Wetherbe
Communications of the ACM November 1989
Volume 32 Issue 11
When a thing is new, people say: "It is not true." Later, when its truth becomes obvious, they say: "It is not important." Finally, when its importance cannot be denied, they say: "Anyway, it is not new." Adapted from William James

36 Multi-level user support through adaptive hypermedia: a highly application-independent help component 80%
L. Miguel Encarnação
Proceedings of the 2nd international conference on Intelligent user interfaces January 1997

37 A geometric modeling and animation system for virtual reality 80%
Mark Green , Sean Halliday
Communications of the ACM May 1996
Volume 39 Issue 5

38 Design compilation of feature-based and constraint-based CAD 80%
Xiangping Chen , Christoph M. Hoffmann
Proceedings of the third ACM symposium on Solid modeling and applications December 1995

39 Integrating simulations with CAD tools for effective facility layout evaluation 80%
Ken Tumay
Proceedings of the 24th conference on Winter simulation December 1992

40 Glide: a language for design information systems 77%
Charles Eastman , Max Henrion
ACM SIGGRAPH Computer Graphics , Proceedings of the 4th annual conference on Computer graphics and interactive techniques July 1977
Volume 11 Issue 2

Results 21 - 40 of 145 **short listing**
Prev Page **1 2 3 4 5 6 7 8** Next Page

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

PORTAL
DIGITAL LIBRARY

> home > about > feedback > login

US Patent & Trademark Office

Try the *new* Portal design
Give us your opinion after using it.

Search Results

Search Results for: **[constraint AND based AND manipulator AND toolset AND editing AND 3d AND objects]**
Found **1** of **126,861 searched.**

Search within Results

GO > Advanced Search

> Search Help/Tips

Sort by: **Title** **Publication** **Publication Date** **Score** Binder

Results 1 - 1 of 1 **short listing**

1 A constraint-based manipulator toolset for editing 3D objects 80%
C. Hsu , G. Alt , Z. Huang , E. Beier , B. Brüderlin
Proceedings of the fourth ACM symposium on Solid modeling and applications
May 1997

Results 1 - 1 of 1 **short listing**

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

PORTAL DIGITAL LIBRARY

> home > about > feedback > login

US Patent & Trademark Office

Try the *new* Portal design

Give us your opinion after using it.

Citation

ACM Symposium on Solid Modeling and Applications >archive

Proceedings of the fourth ACM symposium on Solid modeling and applications >toc

1997 , Atlanta, Georgia, United States

A constraint-based manipulator toolset for editing 3D objects

Authors

C. Hsu
G. Alt
Z. Huang
E. Beier
B. Brüderlin

Sponsor

SIGGRAPH : ACM Special Interest Group on Computer Graphics and Interactive Techniques

Publisher

ACM Press New York, NY, USA

Pages: 168 - 180 Series-Proceeding-Article
Year of Publication: 1997
ISBN:0-89791-946-7

doi> http://doi.acm.org/10.1145/267734.267779 (Use this link to Bookmark this page)

> full text > references > index terms > peer to peer

> Discuss > Similar > Review this Article Save to Binder

> BibTex Format

↑ **FULL TEXT:** Access Rules

pdf 1.71 MB

↑ **REFERENCES**

Note: OCR errors may be found in this Reference List extracted from the full text article. ACM has opted to expose the complete List rather than only correct and linked references.

1 B. Aldefeld, Variation of geometrics based on a geometric-reasoning method, Computer-Aided

Design, v.20 n.3, p.117-126, April, 1988

2 G. Alt. Tweak: A 3d toolkit and modeling system using snapping and constraints. Master's thesis, Department of Computer Science, University of Utah, 1996.

3 R. Anderl , R. Mendgen, Parametric design and its impact on solid modeling applications, Proceedings of the third ACM symposium on Solid modeling and applications, p.1-12, May 17-19, 1995, Salt Lake City, Utah, United States

4 E. Beier. A generic approach to computer graphics. In K. Polthier and C. tlege, editors, Visualization and Mathematics. Springer, 1996. Available from ftp://metallica.prakinf.tuilmenau.de/pub/PROJECTS/GENERIC /springer96.ps.gz.

5 R. Berling and M. Rosendahl. Geometry modeling using dimensional constraints. In 9th International Conference on CAD/CAM, Robotics and Factories of the Future, 1993.

6 E. Bier. Snap dragging in 3d. In Proceedings of the 1986 Workshop on Interactive 3D Graphics, A CM SIGGRAPtt, Chapel Hill, North Carolina, 1986.

7 Alan Borning, The Programming Language Aspects of ThingLab, a Constraint-Oriented Simulation Laboratory, ACM Transactions on Programming Languages and Systems (TOPLAS), v.3 n.4, p.353-387, Oct. 1981

8 W. Bouma, I. Fudos, C.M. Hoffmann, aiazhen Cai, and Robert Paige. A geometric constraint solver. Computer Aided Design, 27(6):487-501, June 1995.

9 Beat Brüderlin, Constructing three-dimensional geometric objects defined by constraints, Proceedings of the 1986 workshop on Interactive 3D graphics, p.111-129, January 1987, Chapel Hill, North Carolina, United States

10 B. Bruderlin. Rule-Based Geometric Modelling. PhD thesis, ETH Zfirich, Switzerland, 1987.

11 Beat Brüderlin, Using geometric rewrite rules for solving geometric problems symbolically, Theoretical Computer Science, v.116 n.2, p.291-303, Aug. 16, 1993

12 Brookshire D. Conner , Scott S. Snibbe , Kenneth P. Herndon , Daniel C. Robbins , Robert C. Zeleznik , Andries van Dam, Three-dimensional widgets, Proceedings of the 1992 symposium on Interactive 3D graphics, p.183-188, June 1992, Cambridge, Massachusetts, United States

13 L. Eggli, C. Hsu, G. Elber, and B. Bruderlin. Inferring 3d models from freehand sketches and constraints. Computer Aided Design, February 1997.

14 Mingxian Fa , Terrence Fernando , Peter M. Dew, Interactive constraint-based solid modeling using allowable motion, Proceedings on the second ACM symposium on Solid modeling and applications, p.243-252, May 19-21, 1993, Montreal, Quebec, Canada

15 C. Hoffmann and P.J. Vermeer. Geometric constraint solving in R2 and R3. In Computing in Euclidean Geometry, 2nd Edition. World Scientific Publishing Co Pte Ltd, 1994.

16 Ching-Yao Hsu, Graph-based approach for solving geometric constraint problems, University of Utah, Salt Lake City, UT, 1996

17 C. Hsu and B. Br'iiderlin. A graph-based degree of freedom analysis algorithm to solve geometric constraint problems. In Proceedings of Theory and Practice of Geometric Modeling (Blaubeuren II), Blaubeuren, October 1996 (An extended version of this paper will be published in a

book by Springer Verlag, 1997).

18 Ching-yao Hsu , Beat Brüderlin, Constraint objects: integrating constraint definition and graphical interaction, Proceedings on the second ACM symposium on Solid modeling and applications, p.467-468, May 19-21, 1993, Montreal, Quebec, Canada

19 C. Hsu and B. Briiderlin. A hybrid geometric constraint solver using exact and iterative geometric constructions. Presented at the Dagstuhl Seminar "CAD-Tools for Products", September 1995, to appear in "CAD Tools and Methods for Design System Development", D. Roller and P. Brunet eds, Springer- Verlag, 1997.

20 V.C. Lin, D.C. Gossard, and R.A. Light. Variational geometry in computer-aided design. Computer Graphics, 15(3):171-177, August 1981.

21 Gregory M. Nielson , Dan R. Olsen, Jr., Direct manipulation techniques for 3D objects using 2D locator devices, Proceedings of the 1986 workshop on Interactive 3D graphics, p.175-182, January 1987, Chapel Hill, North Carolina, United States

22 J. C. Owen, Algebraic solution for geometry from dimensional constraints, Proceedings of the first ACM symposium on Solid modeling foundations and CAD/CAM applications, p.397-407, June 05-07, 1991, Austin, Texas, United States

23 A. Rappoport. Direct manipulation devices for the design of geometric constraint networks. In Magnenat-Thalmann N. and Thalmann D., editors, Communicating with Virtual Worlds - Proceedings Computer Graphics International, Lausanne. Springer Verlag, 1993.

24 Ari Rappoport , Maarten Van Emmerik, User-interface devices for rapid and exact number specification, ACM Transactions on Graphics (TOG), v.12 n.4, p.348-354, Oct. 1993

25 D. Roller, An approach to computer-aided parametric, Computer-Aided Design, v.23 n.5, p.385-391, June 1991

26 Jaroslaw R. Rossignac, Constraints in constructive solid geometry, Proceedings of the 1986 workshop on Interactive 3D graphics, p.93-110, January 1987, Chapel Hill, North Carolina, United States

27 D. Serrano and D.C. Gossard. Combining mathematical models and geometric models in CAE systeins. In Proc. ASME Computers in Eng. Conf., pages 277-284, Chicago, July 1986.

28 W. Sohrt and B.D. Briiderlin. Interaction with constraints in 3D modeling. International Journal of Computational Geometry and Application, 1(4):405-425, December 1991.

29 L. Solano and P. Brunet. Constructive constraintbased model for parametric cad systems. Computer Aided Design, 26(8), 1994.

30 I. Sutherland. Sketchpad, a man-machine graphical communication system. PhD thesis, MIT, January 1963.

31 Robert C. Zeleznik , Kenneth P. Herndon , Daniel C. Robbins , Nate Huang , Tom Meyer , Noah Parker , John F. Hughes, An interactive 3D toolkit for constructing 3D widgets, Proceedings of the 20th annual conference on Computer graphics and interactive techniques, p.81-84, September 1993

↑ **INDEX TERMS**

Primary Classification:
J. Computer Applications

Additional Classification:
F. Theory of Computation
↳ **F.2** ANALYSIS OF ALGORITHMS AND PROBLEM COMPLEXITY
↳ **F.2.2** Nonnumerical Algorithms and Problems
↳ **Subjects:** Geometrical problems and computations

G. Mathematics of Computing
↳ **G.2** DISCRETE MATHEMATICS

I. Computing Methodologies
↳ **I.2** ARTIFICIAL INTELLIGENCE
↳ **I.2.9** Robotics
↳ **Subjects:** Manipulators
↳ **I.3** COMPUTER GRAPHICS
↳ **I.3.4** Graphics Utilities
↳ **Subjects:** Graphics editors

General Terms:
Design, Measurement, Performance, Theory

↑ **Peer to Peer - Readers of this Article have also read:**

- Inferring constraints from multiple snapshots
 ACM Transactions on Graphics (TOG) 12, 4
 David Kurlander , Steven Feiner

- Data structures for quadtree approximation and compression
 Communications of the ACM 28, 9
 Hanan Samet

- The state of the art in automating usability evaluation of user interfaces
 ACM Computing Surveys (CSUR) 33, 4
 Melody Y. Ivory , Marti A Hearst

- Efficient techniques for interactive texture placement
 Proceedings of the 21st annual conference on Computer graphics and interactive techniques
 Peter Litwinowicz , Gavin Miller

- A lifecycle process for the effective reuse of commercial off-the-shelf (COTS) software
 Proceedings of the 1999 symposium on Software reusability
 Christine L. Braun

The ACM Portal is published by the Association for Computing Machinery. C pyright © 2004 ACM, Inc.

A Constraint-based Manipulator Toolset for Editing 3D Objects

C. Hsu‡, G. Alt‡, Z. Huang‡, E. Beier† , B. Brüderlin†
‡Computer Science Department, University of Utah
†Technical University of Ilmenau, PF 0565, D-98684 Ilmenau, Germany

Abstract

We present TWEAK, a new constraint-based manipulator tool set for editing CAD models. Manipulators for positioning vertices, translating and orienting planes, and for translating and orienting rigid objects, appear on the graphic elements that are picked by the interactive user. These manipulators may be controlled by dragging handles. They are connected to an efficient three-dimensional geometric constraint solver, which ensures that the changes are consistent with the relationships defined between the geometric elements of the model. The manipulators can also be used to snap together objects to build assemblies, and they serve as temporary constraints to explicitly control the behavior of the constraint system in underconstrained situations. The manipulators also provide an optional grid system, discretizing the transformations and dimensions. The multiple functionality of the manipulators makes modifications of CAD models easy and intuitive, and at the same time, they provide the necessary accuracy. A few mouse clicks and drag operations can achieve what previously required a complex sequence of operations in CAD systems.

1 Introduction

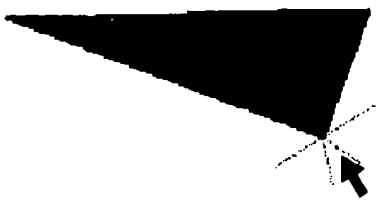
The direct manipulation paradigm is now applied commonly in 2 D illustration and 3D computer animation applications but it is still rarely used in 3D Computer Aided Design. Nevertheless, the need for intuitive tools with a simple point and click paradigm arises also for CAD applications, especially in the conceptual design phase. While in illustration and animation applications the need for accuracy is often limited to visual appearance, in mechanical CAD, data have to be manipulated at least at floating point precision. This is one reason why, traditionally, exact coordinate input, geometric constructions, procedural and parametric modeling, have been used in mechanical design. While such a traditional approach yields accurate data, it requires a lot of work and planning ahead. Changing the parametrisation of an already specified part, often means to redesign the object from scratch. In conceptual design one would like to start with a rough sketch, capturing the design intent, and then refine the input with exact dimensions, as one goes along. Another major application for so-called non-history-based modeling arises when CAD data stemming from a reverse engineering process, or from a hand sketch, need to be modified. These models do not have a prametrisation scheme defined. Also, geometry data exchanged between CAD systems do not necessarily carry parametric information. Moreover, non-history based modeling is much more flexible than parametric modeling because it lets users manipulate models in a more associative, rather than a prescribed way. The approach we take here, is to combine the methods of direct manipulation with discretized input, constraint solving and snap operations in a unified framework, which makes TWEAK a quite powerful modeling tool.

After describing related research, the functionality of the manipulator set is described in section 3. In section 4, the concept of 3D geometric constraints, and the graph representation used here to represent them, are illustrated. We describe the concept of typed degrees of freedom, and local constraints, and give examples how these can be used to control the behavior of the constraint- based manipulation intuitively, especially in under-constrained situations. Section 5 introduces snap-dragging operations which are also integrated into the constraint system. Section 6 describes the object-oriented aspects of the software design.

Permission to make digital/hard copies of all or part of this material for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copyright is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires specific permission and/or fee
Solid Modeling '97 Atlanta GA USA
Copyright 1997 ACM 0-89791-946-7/97/05 ..$3.50

2 Related Work

A few researchers investigated the use of 'direct ma-



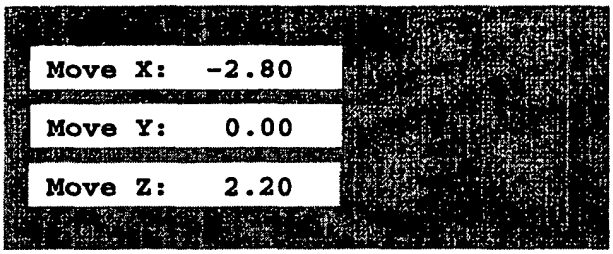


Figure 1: Dragging a point in two directions, using a vertex handle.

nipulation devices' (e.g. the Brown University widget set [31] or the SGI Explorer library [12], as well as the direct manipulation devices described in [23] or [21]). Those applications do not yet handle geometric constraints. The article [24] describes a widget set to interact with a 2D constraint system. Snap-dragging in 3D was proposed by Bier [6]. In [28] we describe an integration between snap-dragging and 2D constraints as well as some 3D degree of freedom analysis. In [14] the use of datagloves for three-dimensional placing of parts in assemblies is described. In the paper presented here, we extend these previous approaches, by integrating the snap-dragging approach with a new 3D constraint solver.

Much research in constraint solving has been conducted in the last years, mostly in the 2D domain [1, 5, 7, 10, 11, 8, 18, 19, 16, 20, 22, 25, 27, 29, 30], but also some 3D [9, 13, 17, 26, 28, 15]. A survey on constraint applications in CAD can be found in [3].

3 Manipulators

A number of manipulators are provided by TWEAK to manipulate vertices or faces of an object, and to orient and position rigid objects. Figure 1 shows the example of a vertex of a tetrahedron, picked by the user. A three dimensional crosshair is shown which acts as a handle to position the picked vertex interactively by dragging the mouse in the direction of one of the coordinate axis. In

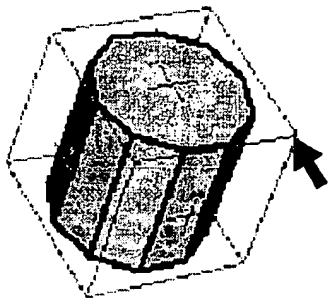


Figure 2: Translating and rotating an object.

the example of a tetrahedron, such translation of vertex points will update the adjacent faces correctly without any special measures since these faces are simplices. In general, however, it will be necessary to employ a constraint solver to ensure the consistent modification of the shape. Even in the case of a simple cube, dragging one of the vertices in any direction would make some faces non-planar. The constraint-based manipulation will be explained in the next section.

A face manipulation handle may be placed on a selected face (exactly at the point inside the face where it was picked) pointing in the direction of the face normal (see figure 10). The orientation of the selected plane can be altered by picking the top end of the handle. The face can be rotated in two directions, around the base point of the handle, by manipulating the handle, similar to a joy stick. By picking the bottom and dragging along the line of the handle, a translation in the direction of the surface normal is effected.

We select a whole object (such as the cylinder in fig 2) by double clicking on any part of the object. This causes a handle frame to appear on the selected object. The frame is centered at the origin of the local coordinate system, and oriented along the coordinate axis, and it also has a size slightly larger than the largest extent of the bounding box in all three directions. The frame lets the user orient the object in all directions (like a space ball) by picking any corner point, but also restricted rotation about one selected axis is possible by picking any line parallel to that axis, and dragging orthogonal to that line (like a rolling cylinder). Translation along any axis or within a plane is also possible (for implementational details see [2].)

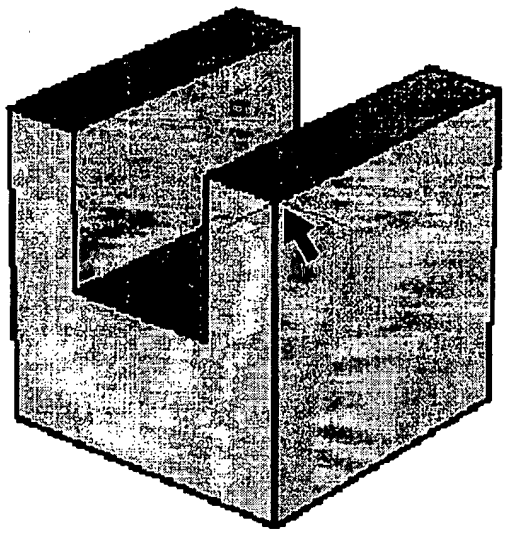


Figure 3: Picking a point to drag.

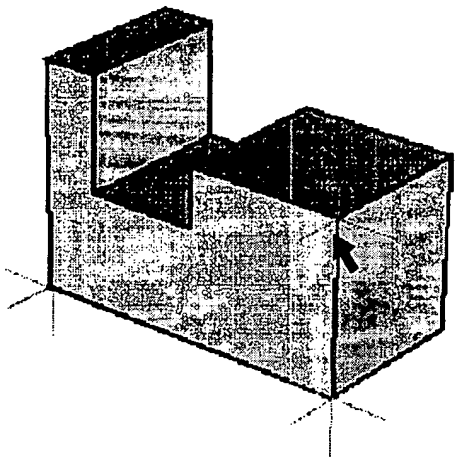


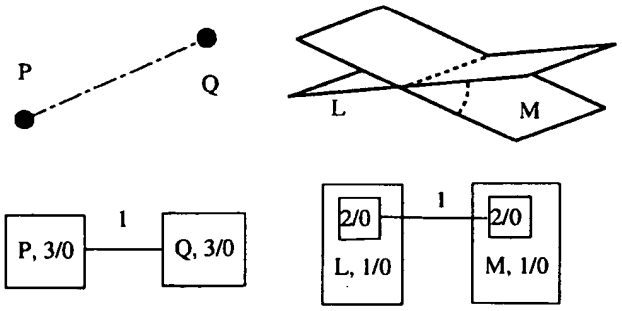
Figure 4: Dragging the picked vertex in three directions – adjacent faces follow because of constraints.

3.1 Discretizing transformations

Any rotation or translation input through the handles described above is discretized by a grid. The numeric values for the transformations are always displayed, as the user drags the handles. The default for angle discretization is five degrees, and the translations are in steps of 0.1 relative to a primitive size of 1.0. (all data is defined and displayed in local coordinates of the objects) This way exact input is possible even when visually manipulating an object or a geometric element within an object. The visual (real-time) feedback lets the user do a cursory placement and once the transformation is approximately right, looking at the numbers allows the user to do an exact input with the same stroke (see fig. 1). Certainly, the grid size can be changed, and the grid may also be turned off.

4 Geometric Constraints

As mentioned previously, data consistency when manipulating objects is not guaranteed by the boundary representation alone, although a polyhedral data structure already implies that all vertices within the same plane are to be coplanar. A constraint solver is employed to maintain the consistency. In figure 3 we show the example of a box with a slot. At this point we impose no explicit constraints (e.g. right angles between adjacent planes), however, the coplanarity of the vertices in each face is imposed automatically, by the constraint solver integrated into the system. When picking the front right point, shown in figure 3, we can move it freely in all three directions. To ensure the planarity of the three faces the constraint system will decide to translate the faces adjacent to the picked point (the result is shown in figure 4). In this section, we describe the representation of constraints, and their evalutation by construction operations.



(a) A distance constraint.

(b) An angle constraint.

Figure 5: Constraining the distance between two points and the angle between planes.

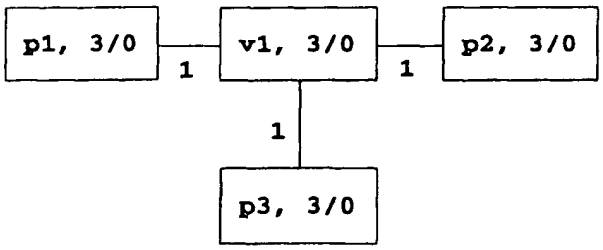


Figure 6: A constraint network representing the incidence relations between a vertex and three planes.

4.1 Graph representation of constraints

The geometric constraints are represented by a graph data structure. In this data structure each geometric element (face, edge, vertex) is represented by a node, and a relationship (constraint) between two or more elements is represented by an arc in the graph (see figure 5). Each geometric element (node) has certain intrinsic degrees of freedom. A vertex, for instance, has three degrees of freedom, which are usually represented by the x/y/z coordinates. A plane also has three degrees of freedom (two angular degrees of freedom, representing the orientation of the plane, and one translational degree of freedom, representing the position). A geometric relationship (constraint) eliminates degrees of freedom between the elements connected by the constraint. The amount of reduction of degrees of freedom by a constraint is called the valency of the constraint [1]. For instance, a parallelism constraint between two planes eliminates 2 degrees of freedom between the planes (i.e. two parallel planes have a total of 4 degrees of freedom as opposed to 6 for two non-parallel planes – the position and two orientation degrees of one plane, and the distance to the second plane). Therefore, a 'parallel' constraint between planes has a valency of 2.

A polygon in the so-called boundary representation can also be represented as a constraint network. Each vertex in a face is related to that face by an incidence relation. A vertex-plane incidence is a valency 1 constraint (instead of 3 degrees of freedoms, the vertex is limited to the 2 dimensional plane, or vice versa, the plane can still be rotated in two directions about the point but loses its translational degree of freedom). By means of the incidence relationship each vertex is incident on at least three planes (see figure ??), and all the vertices of the same face are constrained to be coplanar. Other constraints supported by the system are parallel distance (2 planes are parallel with a fixed distance: valency = 3), fixed angle between two planes (valency = 1) and fixed distance between 2 points (valency = 1).

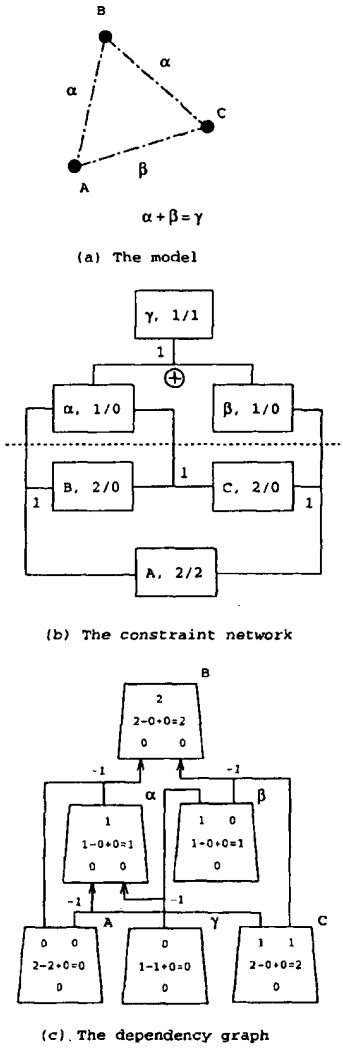
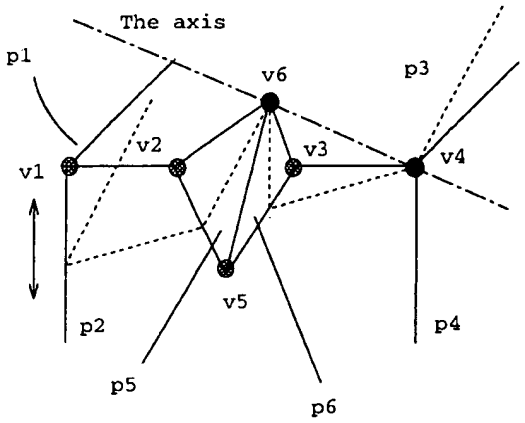


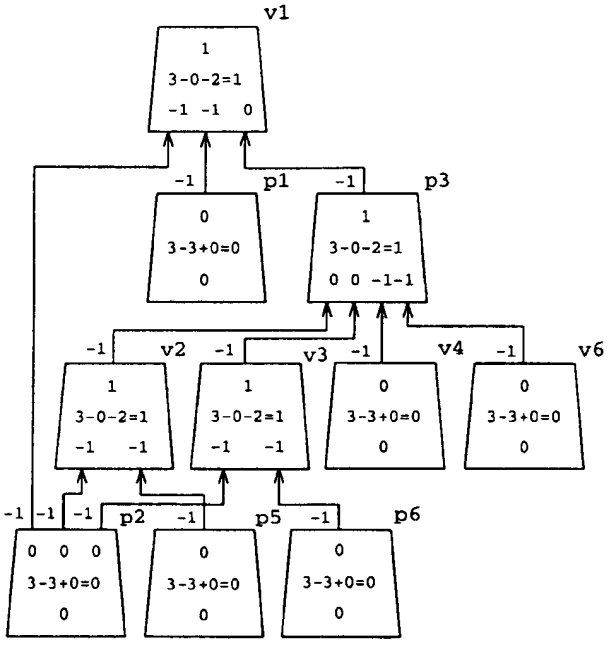
Figure 7: Mixed algebraic and geometric constraints. (a) A geometric model. (b) The constraint graph (c) A dependency graph for dragging point B.

4.2 Evaluating the constraint definitions in interactive situations

When selecting a geometric element for dragging,this action is interpreted by the constraint solver as requesting a certain numer of degrees of freedom from this element in the constraint network. For instance, repositioning a point in space means to request three degrees of freedom from that point; translating a plane means to request one degree of freedom, etc. In [17] we describe how such a request can be met by the constraints, by propagating the request through the constraint network. The paper describes an efficient degree of freedom analysis algorithm which traverses the graph and finds a minimal subgraph of geometric elements that need to be updated as a consequence of the intended manipulation (by recursively requesting degrees of freedom from

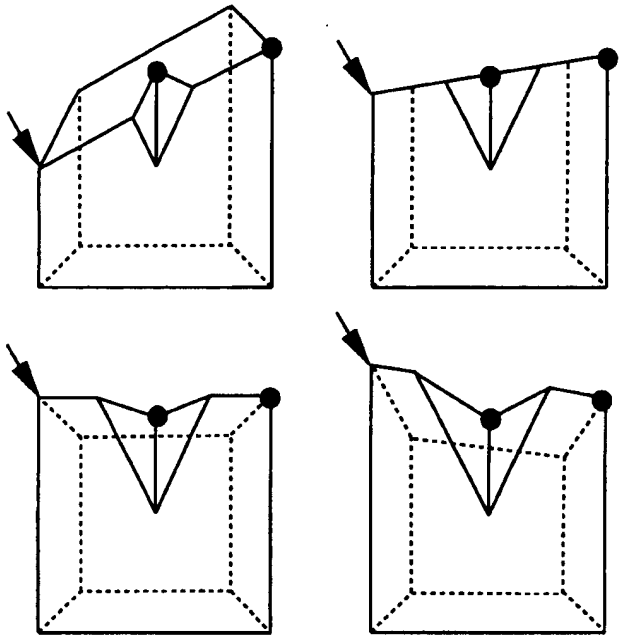


(a)

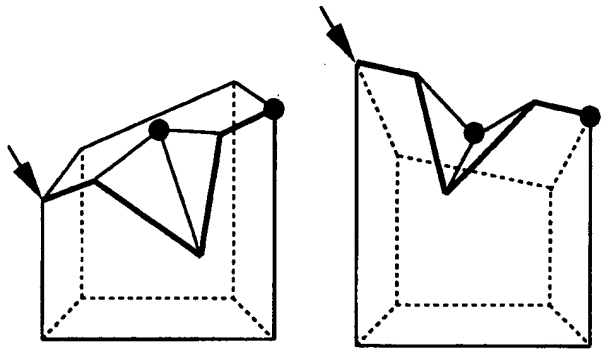


(b)

Figure 8: Editing a 3D polyhedron. (a) Moving vertex v1 along an edge (b)The dependency graph for dragging point v1.



(a)



(b)

Figure 9: Screen shots of the example. (a) Without additional constraints (b) With point - point distance constraints (thick lines).

connected elements). The algorithm is based on the principle of conservation of degrees of freedom, which is expressed by the so-called balance equation for degrees of freedom for each node and arc. The result of the degree of freedom analysis is a dependency graph. The dependency graph is a directed version of a subset of the original constraint graph. The element from which we request degrees of freedoms becomes the root of the dependency graph. The dependency graph is then evaluated by a sequence of construction operations that construct the modified objects, in a top-down order, starting at the root. Those operations are triggered by so-called firing rules. Due to lack of space, we cannot give a detailed description of the degree of freedom analysis and the evaluation by construction operations, in this paper. Instead, we sketch the procedure by examples, below, and refer to [17] and [19] for details.

One of the main advantages of the degree-of-fredom-graph representation of the constraint problem, as well as the degree of freedom analysis algorithm, is that the problem is stated in a very abstract way, independent of the geometric interpretation and the dimension of the embedding space. We can therefore define problems in two and three dimensions, even mixed with algebraic equations between parameters (so-called engineering equations) with one unified framework. On the other hand, the required geometric constructions and algebraic operations necessary to evaluate the dependency graphs mean the implementation of a large number of functions with this approach. In principle, all operations typically occurring in CAD will be used. In our prototype implementation, we limited ourselves to the most common constraint types and operations. In the following, we describe some examples of applications.

Figure 7 shows a 2D triangle with variable distances. Two sides are congruent (length α), and the length β of the third side is related to the other two by the equation $\alpha + \beta = \text{const}\ (\gamma)$. These relations are represented by the graph in 7(b). The lengths themselves are represented by one-dimensional parameter objects. The algebraic equation is represented by the '+' relation which is a hyper arc (with valency 1) between the parameters. When requesting two degrees of freedom from point B, a dependency graph as shown in figure 7(c) is derived. The geometric evaluation is as follows: Point A may remain where it is, and point B follows the cursor. Then α is evaluated as the new distance between points A and B. Then β becomes the difference between γ and the new value of α. Then point C is constructed by intersecting two circles with radii α and β, respectively. The example shows how algebraic operations as well as geometric constructions are used in the evaluation.

Figure 8 shows an example of editing a polyhedron. All vertices are shared by several faces. In the constraint graph this fact is represented by incidence relationships between the corresponding planes and points. When requesting one degree of freedom from v1, the dependency graph shown in figure 8(b) is derived. The geometric interpretation is indicated under (a), as follows: Vertices v6 and v4 remain constant, plane p3 therefore rotates around the axis defined by the two points. The evaluation is again top-down, using geometric constructions. First, the new position of v1 (the root node) is determined by projecting the cursor position onto the line defined by intersecting planes p1 and p2 (which remain constant as well). Then p3 is recalculated from v4, v6 and the new v1. Afterwards v2 and v3 are recalculated by intersecting the new p3 with p5 and v6 (which also remain constant). All other vertices that belong to the top plane p3 also have to be recalculated by corresponding intersections (this is not shown on the dependency graph, in order not to clutter the illustration). All other parts of the polygon remain constant. Figure 9(a) shows screen dumps of the example polyhedron.

For polyhedra, the constructions are typically finding a vertex position by intersecting three planes, or finding a plane through three points, etc. Figure 9(b) shows a modified example with additional point - point distance constraints defined. The behavior is, of course, different, due to the additional constraints. Here the calculations also involve intersecting spheres with spheres and planes.

Figures 10 - 12 show what happens when a face in the rectangular slot feature is rotated about one or two axes. The faces are automatically re-intersected to determine the new vertex locations. The result in figure 12 is a good example for an object which could no longer be defined by a simple sweep of a 2D profile. Without the constraint solver provided here, we would basically be forced to redefine the object by Boolean set operations. Using geometric constraints, the modification becomes simple and intuitive.

More on the evaluation of constraints by geometric construction operations can be found in the publication [19]. The examples shown here can be updated at interactive speed on a low-end workstation or PC. We also ran examples involving several hundred constraints at interactive speed [13], and in most practical applications, unless a high-end graphics workstation is used, the rendering seems to be the bottleneck, and not the constraint solving.

4.3 Typed degrees of freedom and temporary local constraints

In a highly underconstrained object, in general, there are infinitely many solutions to a posed constraint problem. Although there is no absolute best answer to such a problem, we expect that an interactive constraint system behaves predictably, when manipulating the selected elements. We apply two main principles to achieve

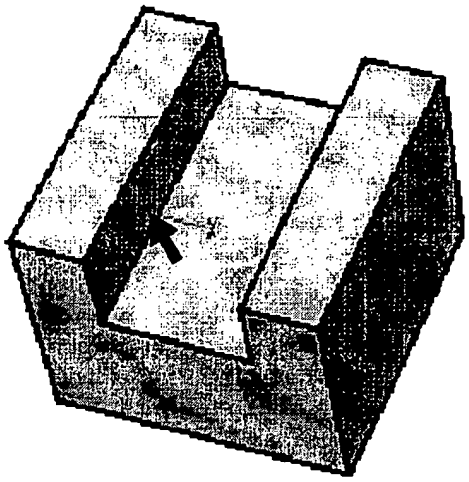


Figure 10: Picking a side of a rectangular slot feature.

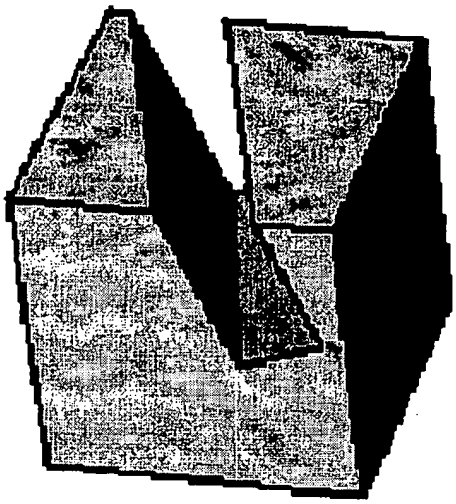


Figure 12: After orienting and scaling the slot feature.

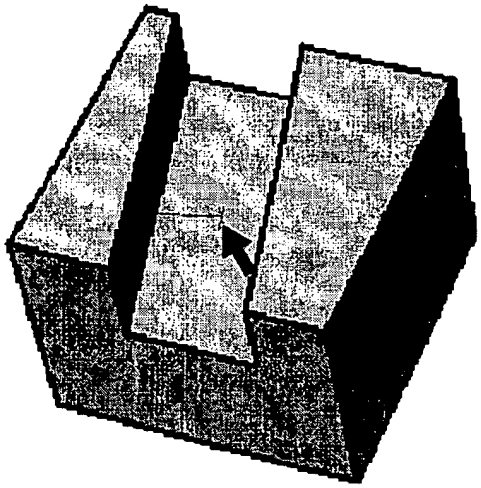


Figure 11: Rotating the face in one direction.

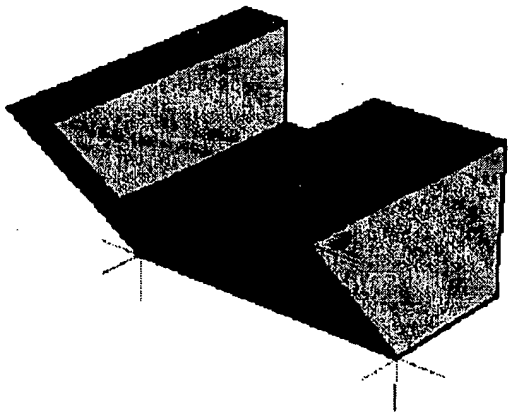


Figure 13: To rotate a face predictably, constrain two points on axis, and drag point forward.

this goal: typed constraints, and temporary constraints.

We distinguish between different types of degrees of freedom (i.e. translational, rotational and dimensional degrees of freedom) and apply a preference order when choosing the possible degrees of freedom in our analysis part of the solution. For instance, it seems obvious that translating a plane has a more predictable outcome than rotating it (rotating can be done about many different axes). Therefore, the translational degrees of freedoms are exploited whenever possible, before other degrees of freedoms are tried. A dimensional degree of freedom (e.g. the distance between two parallel planes) is also preferred over a rotational degree of freedom but less than a translational degree of freedom. In the example of the slot feature, this means that translating one side plane would also translate the opposite side (keeping the width of the slot constant, even without imposing a parallel distance constraint). If, however a vertex of the opposite face has been fixed then translating a vertex of the other side would change the distance between the parallel faces, although a rotation about the point is also a possible solution. Only after explicitly fixing the distance between the faces a rotation will be tried. In [13] we experimented with a mode dependent preference order for the different types of constraints (by introducing so-called soft constraints). Soft constraints allow for even more flexibility, however having different modes can also be confusing.

We also introduce the notion of temporary constraints in our approach. In fact, each handle (which is placed after picking a geometric element) constitutes a local constraint, as long as the handle is not actively manipulated. Local constraints guarantee that object manipulations cannot be inadvertently undone by subsequent manipulations of other elements. As long as a handle is not explicitly deleted (which is simply done by double clicking on the handle) it remains for the rest of the editing session, and fixes all degrees of freedoms of that element. The handle can still be reactivated by dragging it, later on (while the local constraints are suspended). The temporary constraints can also be used to locally control the behavior of the constraint solver in a much more predictable way than a moded constraint hierarchy. Basically, the temporary constraints force the constraint solver to choose a different degree of freedom than it would choose, by default. We show a few examples, below.

In figure 13 we show how the example from figure 4 is further edited. Here, we want to rotate the front face about the bottom edge, while further dragging the same vertex as before. For this purpose, we fix the two vertices at the bottom of the face by picking each, and then dragging the top right vertex, again. The two vertex handles at the bottom act as temporary local constraints and force the constraint solver to rotate the plane around the axis defined by the two points at the bottom. In figure 14, we also fix two vertices at the back side of the object and then drag the top right handle upwards. The values displayed indicate that the dragged vertex was moved by 2.4 units in y, and by 2.0 units in z. It is also possible to rotate a face by a specified angle, about the same axis. Instead of picking and dragging a point in the face, we would pick the face and change its orientation by a specified angle in one direction.

We could easily imagine to extend the idea of using temporary local constraints, involving lines. E.g. to drag a vertex with one degree of freedom, along a line, we would simply pick the line first, and then pick the vertex, and drag it along the line. Again the line would be constrained by a temporary local constraint; therefore, the vertex would only have one degree of freedom left.

5 Snap-dragging and constraints

In this section we show how snap-dragging can be naturally integrated with the constraint solver. We extend on previous work [28] in two ways: A transformation becomes a node in the constraint system. Snapping two objects together establishes a relationship between these objects which is represented by a transformation constraint. Each vertex- or face handle represents a 3D coordinate system. By selecting 2 handles, a transformation between the two elements is defined, which is used to snap the objects together. We will illustrate this with examples.

Figure 15 shows two objects (a cylinder with a slanted side and a box). We select two faces, one on each object, and then snap the objects together by making the handles coincident. The result is shown in figure 16. The block (which is chosen first, and is called the origin) is snapped to the slanted side of the cylinder (the target object). We represent the relationship between the object by a hierarchical transformation data structure which makes the block a dependent (child) of the cylinder(target); i.e. the transformations of the block relative to the cylinder will be automatically restricted to translations within the common plane and rotations along an axis perpendicular to the common plane (figure 17). All transformations performed on the cylinder are also performed on the block.

Different objects are considered rigid relative to each other (i.e. the shape of one object is not affected by shape manipulations of the other). This is realized by representing different objects in different local coordinate systems and by separate constraint networks. However, it is still possible to manipulate the common plane in the target object, which will cause a rigid body transformation in the box (figure 18). On the other

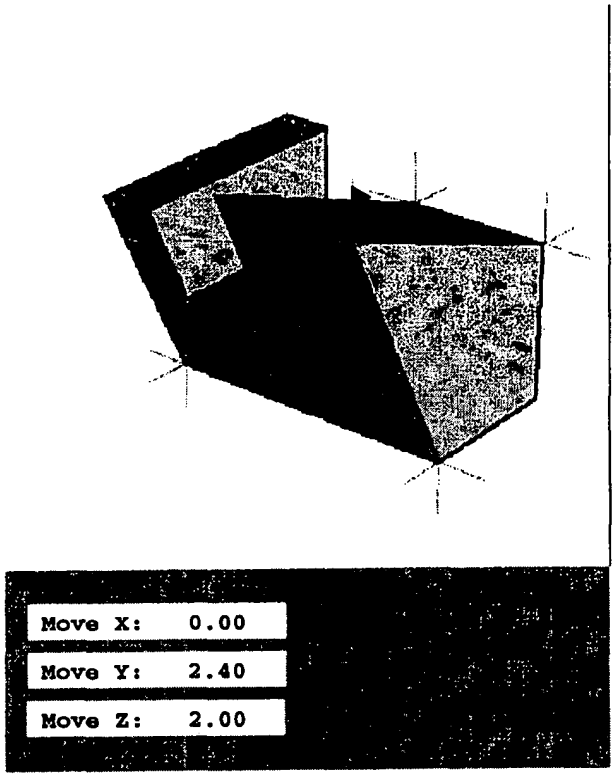


Figure 14: Constrain rotation axis on top face as well, and drag point up.

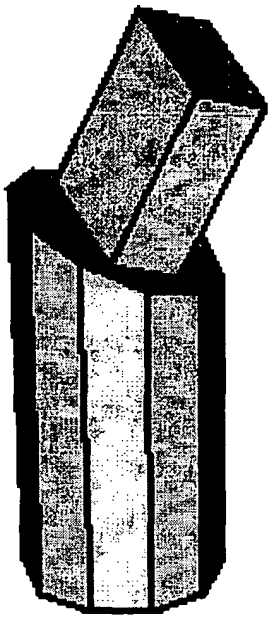


Figure 16: The two objects snapped together.

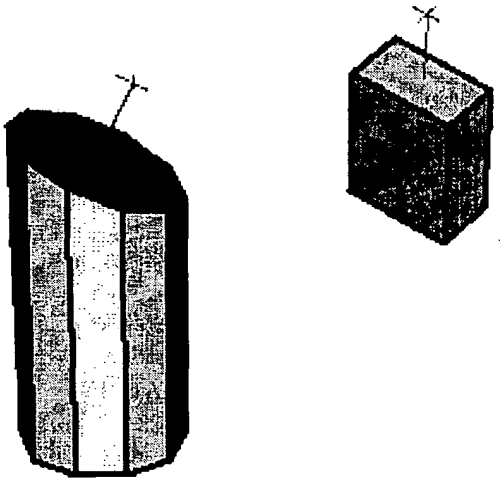


Figure 15: Picking faces on two different objects for snapping.

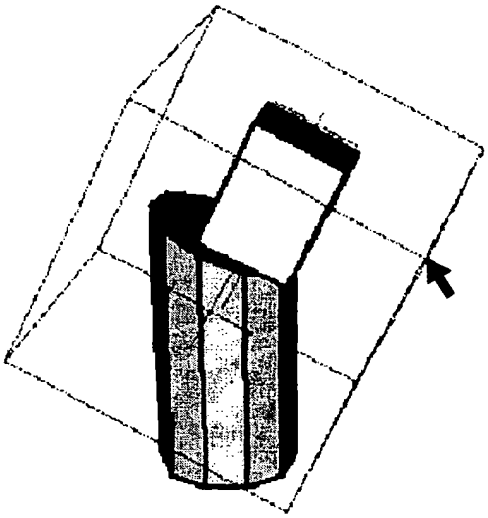


Figure 17: Rotating and translating the box in the common plane.

hand, the bottom plane of the box is fixed within its local coordinate system.

In the hierarchical data structure, the relationship between a node and its logical parent node is further broken up by means of intermediate transformations to accommodate snap-dragging with constraints (see figure 21). Before snapping, the 'Local' transformation transforms the local origin of the child object into the origin of the handle coordinate system. The 'Snap' transformation represents the transformation of the origin handle onto the target handle. This way, all translation and rotation operations for the child object are relative to the target handle, which requires very simple constraint definitions, by just disallowing certain transformations, such as rotating about axes other than z, and translations in the z-direction. After the snap, the target handle is no longer constrained; however, it is still a free node in the constraint network, connected to the element it belongs to. This way, when the target element is updated, the object which snapped to the corresponding handle will be transfomed accordingly (as shown in figure 18). Separating translation and rotations into two nodes makes the two operations commutative (i.e. they are always applied in the same order). Figures 19 - 20 show how a simple effector arm can be assembled by face-face snapping and point-face snapping, which automatically creates the dependencies and constraints simulating cylindric joints and ball joints.

The approach of treating different objects as rigid, relative to each other, has practical advantages in that it eliminates many degrees of freedoms between the objects, and makes the behavior more predictable. If more flexibility is required, this can still be done by merging the objects into one coordinate system, where no assumption about object rigidity is made.

6 Implementation

For the implementation of TWEAK object-oriented concepts are applied.

For instance, the class *Geom3d* defines the following common behavior for all geometrical representations:

- hierarchical modeling with various internal nodes in order to support local modeling and snapping,
- type and bounding box information,
- visibility access,
- interfaces to viewers and interactors, and
- an interface to provide vertices, edges and facets as needed by the interaction handles.

Some of these methods are fully implemented by *Geom3d*, others have to be defined by concrete subtypes. In this way, nearly arbitrary topologies may be handled in a uniform style.

Subtypes of *Geom3d* are, for instance, *Polyhedron* and *BRep*. These types may directly be used to capture the semantic of the application data. Additionally, they provide empty initializers for usage in derived classes. For instance, a subclass of *Polyhedron* that reads polygonal OFF files uses that empty constructor and incrementally creates the data by reading the file.

The implementation of the 3D graphics and interaction component of TWEAK is based on the generic 3D graphics kernel called 'G' [4] which is publically available from the Technical University of Ilmenau. G offers a set of basic types for the major aspects of 3D systems, which may be integrated adaptively by a derived graphics kernel such as TWEAK. On the lowest level, G defines a rendering, interaction, and windowing interface that has been ported to various shading libraries such as OpenGL and Direct-3D, and different windowing systems/ user interface toolkits, for instance, OSF/Motif, Tk, Delphi, MFC.

On the viewer side, polymorphism and extensibility are provided, too. An abstract class *Viewer* defines interfaces that may be implemented in different kind. Concrete subtypes of *Viewer* are an interactive, real-time viewer based on a local illumination model, and a ray-tracing renderer.

This way, TWEAK becomes independent from concrete graphics libraries provided by the environment and, therefore, may be embedded in arbitrary systems.

G and TWEAK are programmed in ANSI-C++ whereby older C++ dialects are supported.

7 Conclusion

TWEAK is an object-oriented, constraint-based toolkit which, we believe, makes direct manipulation feasible for CAD. The approach taken is a combination of direct mouse-based manipulation, a graph-based constraint solver, snap-dragging, and grid input. This is made possible by the availability of a very efficient constraint solver which is capable of solving geometric constraints in two and three dimensions as well as simple algebraic relationships in a single framework.

TWEAK combines the ease of use of direct manipulation with the accuracy requirements of such applications as CAD and conceptual design. We should emphasize that all the examples of interaction described here were possible with one single mode (no menu interaction is necessary, but only simple point and click, double click and dragging with a single button mouse). This is possible mainly through the multiple functionality of the handles which are used for picking and manipulating objects, as well as for snap-dragging and as local constraints.

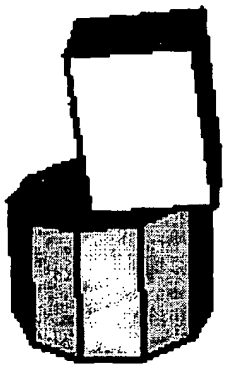


Figure 18: Manipulating the reference plane of the target object re-orients the dependent box.

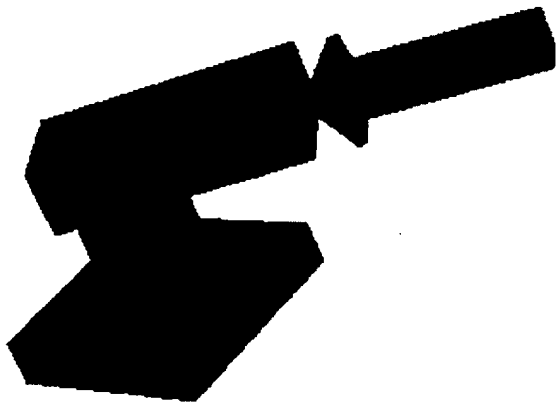


Figure 19: Snapping together several primitives.

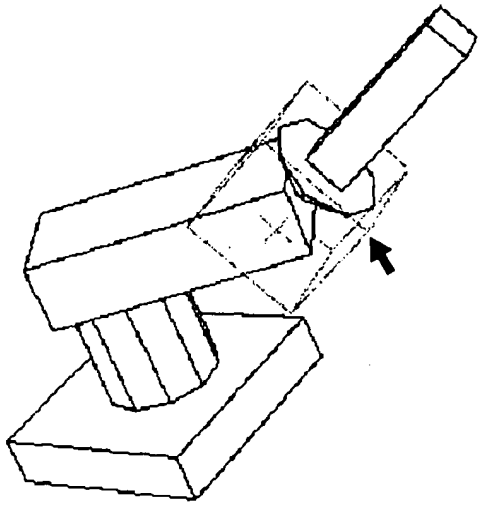


Figure 20: Creating a ball joint through a point-face snap.

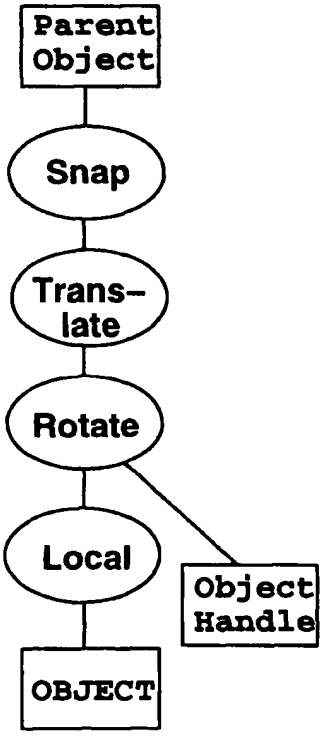


Figure 21: Representing a parent-child relationship through intermediate transformation nodes.

The added convenience offered by using a constraint solver has the promise of shortening the design process significantly. Even the simple examples shown in this paper would currently require much more complex definitions by auxiliary constructions and Boolean set operations. The editing operations proposed here can be used, for instance, for editing objects that come from a sketch input (such as described in [13]) but also in combination with conventional CAD operations (e.g. sweep or CSG) or to edit objects generated by reverse engineering.

It becomes clear that the added flexibility which a constraint-based approach provides over a parametric approach demands new modeling paradigms. In a parametric definition, changing a parameter value, in most cases, has an unambiguous outcome. Since geometric constraints are bi-directional there are many different ways of interpreting such a change, especially in underconstrained situations. In this paper we introduced a few concepts to disambiguate the many choices. This is done by temporary local constraints (which give more control to the user) a preference order between degrees of freedoms, and by the default assumption that objects in different coordinate systems are rigid with respect to one another. We anticipate, however, that many more new paradigms will have to be developed to fully use the potential provided by constraint solving.

8 Acknowledgments

This work was supported in part by the Electronics and Telecommunications Research Institute of Korea, and by Conceptual Geometry LLC.

References

[1] B. Aldefeld. Variation of geometries based on ageometric-reasoning method. *Computer Aided Design*, 20(3):117–126, April 1988.

[2] G. Alt. Tweak: A 3d toolkit and modeling system using snapping and constraints. Master's thesis, Department of Computer Science, University of Utah, 1996.

[3] R. Anderl and R. Mendgen. Parametric design and its impact on solid modeling applications. In *Proceedings of the Third Symposium on Solid Modeling and Applications*, Salt Lake City, 1995. ACM Press.

[4] E. Beier. A generic approach to computer graphics. In K. Polthier and C. Hege, editors, *Visualization and Mathematics*. Springer, 1996. Available from ftp://metallica.prakinf.tu-ilmenau.de/pub/PROJECTS/GENERIC /springer96.ps.gz.

[5] R. Berling and M. Rosendahl. Geometry modeling using dimensional constraints. In *9th International Conference on CAD/CAM, Robotics and Factories of the Future*, 1993.

[6] E. Bier. Snap dragging in 3d. In *Proceedings of the 1986 Workshop on Interactive 3D Graphics, ACM SIGGRAPH*, Chapel Hill, North Carolina, 1986.

[7] A. H. Borning. The programming language aspects of ThingLab, a constraint-oriented simulation laboratory. *ACM Transactions on Programming Languages and Systems*, 3(4):353–387, October 1981.

[8] W. Bouma, I. Fudos, C.M. Hoffmann, Jiazhen Cai, and Robert Paige. A geometric constraint solver. *Computer Aided Design*, 27(6):487–501, June 1995.

[9] B. Bruderlin. Constructing three-dimensional geometric object defined by constraints. In *Proceedings of the 1986 Workshop on Interactive 3D Graphics, ACM SIGGRAPH*, Chapel Hill, North Carolina, 1986.

[10] B. Bruderlin. *Rule-Based Geometric Modelling*. PhD thesis, ETH Zürich, Switzerland, 1987.

[11] B.D. Bruderlin. Using geometric rewrite rules for solving geometric problems symbolically. *Theoretical Computer Science*, 2(116):291–303, August 1993.

[12] D. Brookshire Conner, Scott S. Snibbe, Kenneth P. Herndon, Daniel C. Robbins, Robert C. Zeleznik, and Andries van Dam. Three-dimensional widgets. In David Zeltzer, editor, *Computer Graphics (1992 Symposium on Interactive 3D Graphics)*, volume 25, pages 183–188, March 1992.

[13] L. Eggli, C. Hsu, G. Elber, and B. Bruderlin. Inferring 3d models from freehand sketches and constraints. *Computer Aided Design*, February 1997.

[14] M. Fa, T. Fernando, and P.M. Dew. Interactive constraint-based solid modelling using allowable motion. In *Proceedings of the 1993 ACM/SIGGRAPH Symposium on Solid Modeling Foundations and CAD/CAM Applications*, Montreal, Canada, May 19-21 1993.

[15] C. Hoffmann and P.J. Vermeer. Geometric constraint solving in R^2 and R^3. In *Computing in Euclidean Geometry, 2nd Edition*. World Scientific Publishing Co Pte Ltd, 1994.

[16] C. Hsu. *Graph-Based Approach for Solving Constraint Problems*. PhD thesis, Computer Science Department, University of Utah, May 1996.

[17] C. Hsu and B. Brüderlin. A graph-based degree of freedom analysis algorithm to solve geometric constraint problems. In *Proceedings of Theory and Practice of Geometric Modeling (Blaubeuren II)*, Blaubeuren, October 1996 (An extended version of this paper will be published in a book by Springer Verlag, 1997).

[18] C. Hsu and B. Brüderlin. Constraint objects - integrating constraint definition and graphical interaction. In *Proceedings of the 1993 ACM/SIGGRAPH Symposium on Solid Modeling Foundations and CAD/CAM Applications*, Montreal, Canada, May 19-21 1993.

[19] C. Hsu and B. Brüderlin. A hybrid geometric constraint solver using exact and iterative geometric constructions. *Presented at the Dagstuhl Seminar "CAD-Tools for Products", September 1995, to appear in "CAD Tools and Methods for Design System Development", D. Roller and P. Brunet eds, Springer-Verlag*, 1997.

[20] V.C. Lin, D.C. Gossard, and R.A. Light. Variational geometry in computer-aided design. *Computer Graphics*, 15(3):171–177, August 1981.

[21] G. Nielson and D. Olsen, Jr. Direct manipulation techniques for 3D objects using 3D locator devices. In *Proceedings of the 1986 Symposium on Interactive 3D Graphics*, 1987.

[22] J.C Owen. Algebraic solution for geometry from dimensional constraints. In *Proceedings of the 1991 ACM/SIGGRAPH Symposium on Solid Modeling Foundations and CAD/CAM Applications*, May 1991.

[23] A. Rappoport. Direct manipulation devices for the design of geometric constraint networks. In Magnenat-Thalmann N. and Thalmann D., editors, *Communicating with Virtual Worlds - Proceedings Computer Graphics International, Lausanne*. Springer Verlag, 1993.

[24] A. Rappoport and M.J.G.M van Emmerik. User interface devices for rapid and exact number specification. *ACM Transactions on Graphics*, 12:348–354, 1993.

[25] D. Roller. An approach to computer-aided parametric design. *Computer Aided Design*, 23(5):385–391, June 1991.

[26] Jaroslaw P. Rossignac. Constraints in constructive solid geometry. In *Proceedings of Workshop on Interactive 3D Graphics*, pages 93–110, Chapel Hill, NC, October 23-24 1986.

[27] D. Serrano and D.C. Gossard. Combining mathematical models and geometric models in CAE systems. In *Proc. ASME Computers in Eng. Conf.*, pages 277–284, Chicago, July 1986.

[28] W. Sohrt and B.D. Brüderlin. Interaction with constraints in 3D modeling. *International Journal of Computational Geometry and Application*, 1(4):405–425, December 1991.

[29] L. Solano and P. Brunet. Constructive constraint-based model for parametric cad systems. *Computer Aided Design*, 26(8), 1994.

[30] I. Sutherland. *Sketchpad, a man-machine graphical communication system.* PhD thesis, MIT, January 1963.

[31] Robert C. Zeleznik, Kenneth P. Herndon, Daniel C. Robbins, Nate Huang, Tom Meyer, Noah Parker, and John F. Hughes. An interactive 3D toolkit for constructing 3D widgets. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 81–84, August 1993.

PORTAL
DIGITAL LIBRARY

> home > about > feedback > login

US Patent & Trademark Office

Try the *new* Portal design
Give us your opinion after using it.

Search Results

Search Results for: **[(area and volume)<AND>(((housing or building)<AND>((3D<AND>((CAD and architecture and cost)) <AND>(meta_published_date <= 01-01-1998)))))]**
Found **47** of **126,861 searched.**

Search within Results

GO > Advanced Search

> Search Help/Tips

Sort by: **Title** **Publication** **Publication Date** **Score** Binder

Results 1 - 20 of 47 **short listing**

Prev Page 1 2 3 Next Page

1 The triangle processor and normal vector shader: a VLSI system for high performance graphics 82%
Michael Deering , Stephanie Winner , Bic Schediwy , Chris Duffy , Neil Hunt
ACM SIGGRAPH Computer Graphics , Proceedings of the 15th annual conference on Computer graphics and interactive techniques June 1988
Volume 22 Issue 4

2 Machine interpretation of CAD data for manufacturing applications 82%
Qiang Ji , Michael M. Marefat
ACM Computing Surveys (CSUR) September 1997
Volume 29 Issue 3
Machine interpretation of the shape of a component for CAD databases is an important problem in CAD/CAM, computer vision, and intelligent manufacturing. It can be used in CAD/CAM for evaluation of designs, in computer vision for machine recognition and machine inspection of objects, and in intelligent manufacturing for automating and integrating the link between design and manufacturing. This topic has been an active area of research since the late '70s, and a significant number of computat ...

3 Visibility culling using hierarchical occlusion maps 82%
Hansong Zhang , Dinesh Manocha , Tom Hudson , Kenneth E. Hoff
Proceedings of the 24th annual conference on Computer graphics and interactive techniques August 1997

4 Pen computing: a technology overview and a vision 80%
André Meyer

ACM SIGCHI Bulletin July 1995
Volume 27 Issue 3

This work gives an overview of a new technology that is attracting growing interest in public as well as in the computer industry itself. The visible difference from other technologies is in the use of a pen or pencil as the primary means of interaction between a user and a machine, picking up the familiar pen and paper interface metaphor. From this follows a set of consequences that will be analyzed and put into context with other emerging technologies and visions.Starting with a short historic ...

5 Model-based object recognition in dense-range images—a review 80%

Farshid Arman , J. K. Aggarwal
ACM Computing Surveys (CSUR) March 1993
Volume 25 Issue 1

The goal in computer vision systems is to analyze data collected from the environment and derive an interpretation to complete a specified task. Vision system tasks may be divided into data acquisition, low-level processing, representation, model construction, and matching subtasks. This paper presents a comprehensive survey of model-based vision systems using dense-range images. A comprehensive survey of the recent publications in each subtask pertaining to dense-range image object recogni ...

6 Computer-aided visualization of the human body for industrial designer 80%

Alan Wier
Proceedings of the 18th annual ACM SIGUCCS conference on User services August 1990

7 Accelerated walkthrough of large spline models 80%

Subodh Kumar , Dinesh Manocha , Hansong Zhang , Kenneth E. Hoff
Proceedings of the 1997 symposium on Interactive 3D graphics April 1997

8 HoloSketch: a virtual reality sketching/animation tool 80%

Michael F. Deering
ACM Transactions on Computer-Human Interaction (TOCHI) September 1995
Volume 2 Issue 3

This article describes HoloSketch, a virtual reality-based 3D geometry creation and manipulation tool. HoloSketch is aimed at providing nonprogrammers with an easy-to-use 3D "What-You-See-Is-What-You-Get" environment. Using head-tracked stereo shutter glasses and a desktop CRT display configuration, virtual objects can be created with a 3D wand manipulator directly in front of the user, at very high accuracy and much more rapidly than with traditional 3D drawing systems. HoloSke ...

9 Dimension-independent modeling with simplicial complexes 80%

A. Paoluzzi , F. Bernardini , C. Cattani , V. Ferrucci
ACM Transactions on Graphics (TOG) January 1993
Volume 12 Issue 1

10 Glide: a language for design information systems 77%

Charles Eastman , Max Henrion
ACM SIGGRAPH Computer Graphics , Proceedings of the 4th annual conference on C mputer graphics and interactive techniques July 1977
Volume 11 Issue 2

11 A language for shading and lighting calculations 77%

Pat Hanrahan , Jim Lawson

ACM SIGGRAPH C mputer Graphics , Proceedings of the 17th annual conference on C mputer graphics and interactive techniques September 1990

Volume 24 Issue 4

12 The emerging technology of CAD/CAM 77%

Larry Lichten

Proceedings of the 1984 annual conference of the ACM on The fifth generation challenge January 1984

Computer-Aided Design and Manufacture (CAD/CAM) represents a merging of technological advances in computer hardware and software with pressing needs in manufacturing industries. Integrated manufacturing systems--from computer graphics-aided design through engineering analysis and automated fabrication--are only now beginning to fulfill nearly twenty-five-year old promises of increased production efficiency. This paper summarizes CAD/CAM's evolution and its current state and then describes s ...

13 Modeling polyhedral solids bounded by multi-curved parametric surfaces 77%

Yehuda E. Kalay

Proceedings of the nineteenth design automation conference January 1982

Since the beginning of geometric modeling as a field of CAD a decade ago, the methods for interactive design of solid objects and interactive design of free-formed surfaces (of degree 3 and higher) were developed along parallel but disjoint strands. One led to the development of techniques for representing and manipulating the shape of polyhedral solids bounded mostly by planes, while the other led to the development of techniques for the mathematical representation of curved surfac ...

14 Information technologies for the 1990s: an organizational impact perspective 77%

Detmar W. Straub , James C. Wetherbe

Communications of the ACM November 1989

Volume 32 Issue 11

When a thing is new, people say: "It is not true." Later, when its truth becomes obvious, they say: "It is not important." Finally, when its importance cannot be denied, they say: "Anyway, it is not new." Adapted from William James

15 DVI—a digital multimedia technology 77%

G. David Ripley

Communications of the ACM July 1989

Volume 32 Issue 7

A digital presentation technology that manages anything from text to full-motion video has the potential of expanding the usefulness of personal computers, while rendering them less intimidating.

16 Plenary: biology and medicine: A computational steering model applied to problems in medicine 77%

Christopher R. Johnson , Steven G. Parker

Proceedings of the 1994 ACM/IEEE conference on Supercomputing November 1994

We describe a computational steering model which allows users to interactively change boundary conditions, model geometry, and computational parameters via a graphical

user interface. To replace the typical simulation mode -- in which the researcher manually sets input parameters, computes results, stores data off to disk, visualizes the results via a separate visualization package, then starts again at the beginning -- we have designed software to "close the loop" and allow the visualization to ...

17 Issues in the design of a flexible distributed architecture for supporting persistence and interoperability in collaborative virtual environments 77%

Jason Leigh , Andrew E. Johnson , Thomas A. DeFanti
Proceedings of the 1997 ACM/IEEE conference on Supercomputing (CDROM)
November 1997

CAVERN, the CAVE Research Network, is an alliance of industrial and research institutions equipped with CAVE-based virtual reality hardware and high-performance computing resources, interconnected by high-speed networks, to support collaboration in design, education, engineering, and scientific visualization.CAVERNsoft is the collaborative software backbone for CAVERN. CAVERNsoft uses distributed data stores to manage the wide range of data volumes (from a few bytes to several terabytes) that ar ...

18 Types and persistence in database programming languages 77%

Malcolm P. Atkinson , O. Peter Buneman
ACM Computing Surveys (CSUR) June 1987
Volume 19 Issue 2

Traditionally, the interface between a programming language and a database has either been through a set of relatively low-level subroutine calls, or it has required some form of embedding of one language in another. Recently, the necessity of integrating database and programming language techniques has received some long-overdue recognition. In response, a number of attempts have been made to construct programming languages with completely integrated database management systems. These lang ...

19 The Quadtree and Related Hierarchical Data Structures 77%

Hanan Samet
ACM Computing Surveys (CSUR) June 1984
Volume 16 Issue 2

20 Letters to the editor: A protection model and its implementation in a dataflow system 77%

Lubomir Bic
Communications of the ACM September 1982
Volume 25 Issue 9

A protection model is presented for a general purpose computing system based on tags attached as seals and signatures to values exchanged among processes. A tag attached to a value as a seal does not prevent that value from being propagated to any place within the system; rather, it guarantees that the value and any information derived from it cannot leave the system unless a matching tag is presented. A tag attached to a value as a signature is used by a p ...

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.